

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

**Utilização do Conceito de *Instant Messaging* em
Gestão de Redes**

José António Folha da Silva

Porto, Novembro de 2006

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Utilização do Conceito de *Instant Messaging* em Gestão de Redes

José António Folha da Silva

Licenciado em Engenharia Electrotécnica
pelo Instituto Superior de Engenharia do Porto,
integrado no Instituto Politécnico do Porto.

Dissertação submetida para satisfação
parcial dos requisitos do grau de Mestre em Redes
e Serviços de Comunicação.

Realizada sob a orientação do Professor
Doutor Raul Filipe Teixeira de Oliveira, do
Departamento de Engenharia Electrotécnica e de
Computadores da Faculdade de Engenharia da
Universidade do Porto.

Porto, Novembro de 2006

*Aos meus pais, avós e irmão,
por tudo aquilo que por mim fazem.
À minha noiva Solange,
pela paciência e por ter acreditado em mim.*

Resumo

Quando falamos em gestão de redes IP cada vez mais se verifica a necessidade de acções proactivas que, pelos seus próprios requisitos, levam a que a acção humana passe a dar lugar a acções autónomas com um certo grau de inteligência. Tais acções só são possíveis de implementar se falarmos em Aplicações Autónomas. Se pensarmos nestas aplicações, facilmente nos apercebemos que elas poderão ser utilizadas para gerir proactivamente uma rede IP nos seus diferentes níveis, elementos, recursos, serviços, utilizadores, etc. Pela sua filosofia, e do ponto de vista humano, a gestão fica mais facilitada.

As Aplicações de gestão são usadas para implementar um modelo de informação de gestão comum. Geralmente os alarmes disparados pelas aplicações de gestão são enviados para uma consola de gestão, enviados por e-mail, ou outra forma de comunicação, para o gestor da rede. É certo que os gestores de rede nem sempre têm permanente contacto visual com o monitor, ou então, não estão presentes no local. Estas situações podem levar à criação de atrasos na resolução de problemas.

Aplicações Autónomas deverão ter aqui um papel importante, na medida em que podem ajudar a resolver alguns destes problemas.

Assim, apesar de já existirem bastantes conceitos na abordagem ao problema, esta tese pretende explorar um outro, nomeadamente o da utilização de *Instant Messaging* para a Gestão de Redes, utilizando um modelo global de informação baseado no LDAP e utilizar o protocolo XMPP como protocolo base de comunicação entre aplicações e transporte de conteúdo XML.

Abstract

When we speak in IP Network Management, each more we verify the need of proactive actions that, for its proper requirements, takes place of the human actions, and gives place to Autonomous Applications with a certain degree of intelligence. Such actions are only possible to implement if we talk about Autonomous Applications. If we think about these applications, we easily understand that they could be used to manage an IP Network at its different levels, such as, elements, resources, services, users, etc., in a proactive way. For its philosophy, and because of the human point of view, the management is more facilitated.

The Management Applications are used to implement a common management information model. Generally the alarms fired by the management applications are sent to a management console, sent by email, or another way of communication, to the network manager. It is certain that network managers don't always have permanent visual contact with the screen, or they are away of the scene. These situations can lead to the creation of delays in the resolution of problems.

Autonomous Applications should have here an important role, in the way that they can help solving some of these problems.

Thus, although already exist many concepts in approach of the problem, this thesis intend to explore another one, the use of Instant Messaging in Network Management, using a global model of information based on the LDAP and to use the XMPP protocol as base of communication between applications and transport of XML content.

Résumé

Quand nous parlons de la gestion de réseaux d'IP, nous vérifions de plus en plus la nécessité d'actions proactives, celles-ci par leur propres conditions mènent à que l'action humaine donne place aux applications autonomes avec un certain degré d'intelligence. Ces actions sont seulement possibles de mettre en application si nous parlons des applications autonomes. En pensant à ces applications, nous comprenons facilement qu'elles pourraient être utilisées pour contrôler un réseau d'IP à des différents niveaux, éléments, ressources, services, utilisateurs, etc. Soit pour sa philosophie, soit du point de vue humain, la gestion est plus simplifiée.

Les applications de gestion sont utilisées pour développer un modèle d'information de gestion commune. Généralement les alarmes déclenchées par les applications de gestion sont envoyées à une console de gestion, envoyée par email, ou par une différente manière de communication, au régisseur du réseau. C'est évident, que les régisseurs du réseau n'ont pas toujours le contact visuel avec l'écran, ou même quand ils sont absent. Ces situations peuvent mener à un retard à la résolution des problèmes.

Les applications autonomes ont un rôle important à ces moments, car elles sont capables de résoudre certains de ces problèmes.

Ainsi, malgré l'existence de beaucoup de concepts qui s'approche au problème, cette thèse prévoient encore explorer une autre , nommément l'utilisation des Instant Messaging dans la Gestion du Réseau, en utilisant un modèle global d'information basé sur le LDAP et pour employer le protocole de XMPP comme base de communication entre les applications et le transport du contenu de XML.

Agradecimentos

Agradeço ao Professor Doutor Raul Filipe Teixeira Oliveira a confiança que depositou em mim ao me propor este trabalho e pelo apoio prestado durante a elaboração do mesmo.

Não posso deixar de agradecer ao colega de doutoramento Bruno Filipe Marques, pela forma de como se empenhou na troca de conhecimentos e experiência.

O meu agradecimento ao Doutor Eduardo Luís Cardoso, Coordenador dos Serviços de Informática da Escola Superior de Biotecnologia da Universidade Católica Portuguesa, pelo tempo que me disponibilizou para a realização deste trabalho.

Por último quero expressar a minha gratidão a toda a minha família e noiva Solange, que me apoiaram e incentivaram desde o início.

Conteúdo

Resumo	i
Abstract.....	iii
Résumé	v
Agradecimentos	vii
Abreviaturas	xvii
1 Introdução	1
1.1 Tema	1
1.2 Objectivos	2
1.3 Estrutura do Tese	3
2 Estado da Arte	5
2.1 Definição de Agente Autónomo	5
2.2 Sistemas Multi-Agente – MAS.....	7
2.3 Coordenação	9
2.4 Architecturas de Agentes.....	10
2.4.1 Architecturas Cognitivas	10
2.4.2 Architecturas Reactivas.....	11
2.4.3 Architecturas Híbridas.....	12
2.5 Modelo Tradicional de Agentes FIPA	12
2.5.1 Agente	12
2.5.2 Facilitador.....	13
2.5.3 Sistema de Gestão de Agentes.....	14
2.5.4 Serviço de Transporte de Mensagens	15
2.5.5 Plataforma de Agentes.....	16
2.5.6 Identificação dos Agentes.....	16
2.6 Linguagem de Comunicação de Agentes.....	17

2.6.1	A Linguagem KQML	18
2.6.2	A linguagem FIPA-ACL	21
2.7	Os Agentes na Gestão de Redes	24
2.8	Benefícios e Expectativas dos Sistemas de Agentes.....	25
2.9	Limitações e Desafios dos Sistemas de Agentes	26
2.10	Gestão Centralizada e Distribuída	27
2.11	Conclusões	28
3	Arquitectura de Gestão	31
3.1	Modelo Gestão de Redes	31
3.2	Arquitectura de Gestão	32
3.2.1	Objecto Gerido	32
3.2.2	Sistema de Gestão da Rede.....	32
3.2.3	Linguagens	35
3.3	Modelo de Informação.....	36
3.3.1	Sistema de Informação de Gestão de Redes.....	37
3.3.2	Barramento de Informação de Gestão de Redes.....	38
3.3.3	LDAP – Lightweight Directory Access Protocol	40
3.3.4	Directório de Informação de Gestão de Redes do domínio	42
3.4	Protocolos Comunicação Gestão	43
3.5	Funcionalidades	45
3.6	Vantagens	47
3.7	Conclusões	48
4	Arquitectura de Comunicação do Sistema de Aplicações Autónomas de Gestão.....	49
4.1	Introdução	49
4.2	Modelo de Informação.....	51
4.3	Aplicações de gestão.....	52
4.3.1	Aplicações Autónomas.....	53
4.3.2	Directório de Agentes.....	54
4.3.3	Directório de Serviços	55
4.4	Plataforma de Comunicação	56
4.4.1	Protocolos Comunicação	57
4.4.2	Protocolos de Gestão	59
4.4.3	LDAP.....	59
4.4.4	Instant Messaging.....	60

4.5	Linguagens Comunicação.....	61
4.5.1	Ontologias	61
4.5.2	XML	62
4.5.3	Mensagens	63
4.6	Segurança.....	66
4.6.1	Vírus	66
4.6.2	Mensagens de Spim.....	66
4.6.3	Privacidade	66
4.7	Conclusão	66
5	Protótipo	69
5.1	Ambiente de Gestão Intra-Domínio.....	69
5.1.1	Protótipo de utilização autónomo	72
5.1.2	Protótipo de utilização com Operador de Rede	78
5.2	Ambiente de Gestão Inter Domínios.....	79
5.2.1	Protótipo de utilização Inter-Domínio	80
5.3	Conclusões	82
6	Conclusões	85
6.1	Trabalho Futuro	86
	Bibliografia	89

Lista de Tabelas

Tabela 1 – Identificação de Mensagens KQML.....	18
Tabela 2 – Performativas KQML.....	19
Tabela 3 – Exemplo de mensagem KQML.....	20
Tabela 4 – Parâmetros predefinidos no FIPA-ACL.....	21
Tabela 5 – Performativas FIPA-ACL.....	22
Tabela 6 – Exemplo de mensagem FIPA-ACL.....	23
Tabela 7 – Exemplo de registo no LDAP.....	59
Tabela 8 – Exemplo de Pesquisa e Resultado no LDAP.....	63
Tabela 9 – Exemplo de Mensagem IM.....	64
Tabela 10 – Exemplo de Mensagem de Conteúdo.....	65
Tabela 11 – Pesquisa/Resultado por Agente/Serviço de Utilizadores no LDAP.....	72
Tabela 12 – Envio do alarme via <i>Instant Messaging</i> pelo <i>Dispatcher Agent</i>	73
Tabela 13 – Pesquisa/Resultado por Agente/Serviço de Correio Electrónico no LDAP.....	74
Tabela 14 – Encaminhamento do alarme pelo agente SGU para o agente CorreioElectronico..	74
Tabela 15 – Pesquisa/Resultado por Agente/Serviço de DNS.....	75
Tabela 16 – Encaminhamento do alarme pelo agente CorreioElectronico para o agente DNS..	76
Tabela 17 – Notificação de problema resolvido para o agente SGU.....	77
Tabela 18 – Pesquisa de contacto do administrador.....	78
Tabela 19 – Notificação ao Administrador.....	78
Tabela 20 – Pesquisa/Resultado por Agente/Serviço de <i>Tickets</i> no LDAP.....	81
Tabela 21 – Notificação do Operador.....	81

Lista de Figuras

Figura 1 – Arquitectura Cognitiva.	10
Figura 2 – Arquitectura de Subsunção.	11
Figura 3 – Modelo de referência para Agentes FIPA.....	12
Figura 4 – Modelo de Gestão de Redes.....	31
Figura 5 – Modelo Sistemas de Gestão de Redes.	32
Figura 6 – Sistema de Informação de Redes.	37
Figura 7 – Bus de Informação de Gestão de Redes baseado no LDAP.	38
Figura 8 – Árvore de Directório de Informação de Gestão de Rede.	39
Figura 9 – Exemplo de árvore de informação LDAP.....	41
Figura 10 – Domínio de Gestão da Rede.	43
Figura 11 – Protocolos de Comunicação no Ambiente de Gestão.	44
Figura 12 – Arquitectura Sistema Gestão.....	45
Figura 13 – Dimensões da Gestão.....	46
Figura 14 – Interfaces para protocolos de comunicação.	47
Figura 15 – Dimensões da Gestão.....	50
Figura 16 – Plataforma de Informação.....	50
Figura 17 – Modelo Informação Global LDAP.	51
Figura 18 – Árvore de Gestão Simplificada.	52
Figura 19 – Árvore de Gestão LDAP Relacionada.	53
Figura 20 – Árvore de Informação das Aplicações Autónomas.....	54
Figura 21 – Plataforma de Comunicação.	56
Figura 22 – Comunicação entre Aplicações Autónomas e LMIB.....	57
Figura 23 – Comunicação entre Aplicações Autónomas e Utilizadores.	58
Figura 24 – Comunicação entre Utilizadores e LMIB.	58

Figura 25 – Ontologia com relação Área funcional com Níveis da Gestão.	62
Figura 26 – Ambiente de rede Intra-Domínio.	70
Figura 27 – Ambiente de Gestão.	70
Figura 28 – Árvore de Informação de Gestão.	71
Figura 29 – Protótipo Inter Domínio 1.	72
Figura 30 – Protótipo Inter Domínio 2.	74
Figura 31 – Protótipo Inter Domínio 3.	75
Figura 32 – Protótipo Inter Domínio 4.	77
Figura 33 – Protótipo Inter Domínio com Administrador.	78
Figura 34 – Ambiente de Gestão Inter Domínio.	79
Figura 35 – Protótipo Intra-Domínio.	81

Nomenclatura

Abreviaturas

ACC	–	<i>Agent Communication Channel</i>
ACL	–	<i>Agent Communication Language</i>
AID	–	<i>Agent Identifier</i>
AMS	–	<i>Agent Management System</i>
AP	–	<i>Agent Platform</i>
API	–	<i>Application Programming Interface</i>
ASN.1	–	<i>Abstract Syntax Notation One</i>
CBL	–	<i>Common Business Library</i>
CMIP	–	<i>Common Management Information Protocol</i>
CN	–	<i>Common Name</i>
DF	–	<i>Directory Facilitator</i>
DIT	–	<i>Directory Information Tree</i>
DN	–	<i>Distinguished Name</i>
DNS	–	<i>Domain Name System</i>
DPS	–	<i>Distributed Problem Solving</i>
DTD	–	<i>Document Type Definition</i>
FIPA	–	<i>Foundation for Intelligent and Physical Agents</i>
FQDN	–	<i>Fully Qualified Domain Name</i>
FTP	–	<i>File Transfer Protocol</i>
HAP	–	<i>Home Agent Platform</i>
HTML	–	<i>Hyper Text Markup Language</i>
IDL	–	<i>Interface Definition Language</i>
IETF	–	<i>Internet Engineering Task Force</i>
IMAP	–	<i>Internet Message Access Protocol</i>

IRC	–	<i>Internet Relay Chat</i>
KIF	–	<i>Knowledge Interchange Format</i>
LDAP	–	<i>Lightweight Directory Access Protocol</i>
LMIB	–	<i>LDAP Management Information BUS</i>
MANIA	–	<i>Managing Awareness Networks Through Intelligent Agents</i>
MAS	–	<i>Multi Agent Architecture</i>
MIB	–	<i>Management Information Base</i>
MTP	–	<i>Message Transport Protocol</i>
MTS	–	<i>Message Transport Service</i>
NAMS	–	<i>Network Asset Management System</i>
NFS	–	<i>Network File System</i>
NMIB	–	<i>Network Management Information Bus</i>
NMID	–	<i>Network Management Information Domain</i>
NMIS	–	<i>Network Management Information System</i>
NMS	–	<i>Network Management System</i>
NOSS	–	<i>Network Operations Support System</i>
NUMS	–	<i>Network Users Management System</i>
OIL	–	<i>Ontology Interchange Language</i>
OMG	–	<i>Object Management Group</i>
PDU	–	<i>Power Data Unit</i>
RDN	–	<i>Relative Distinguished Name</i>
RFC	–	<i>Request For Comment</i>
SGML	–	<i>Standardized Generalized Markup Language</i>
SGU	–	Sistema de Gestão de Utilizadores
SIMPLE	–	<i>SIP Instant Messaging Presence Leveraging Extensions</i>
SLA	–	<i>Service Level Agreement</i>
SMB	–	<i>Server Message Blocks</i>
SMI	–	<i>Structured Management Information</i>
SMTP	–	<i>Simple Mail Transfer Protocol</i>
SNMP	–	<i>Simple Network Management Protocol</i>
SPAM	–	Mensagem Electrónica não Solicitada enviada em Massa
SPIM	–	<i>SPAM em Instant Messaging</i>
TI	–	Tecnologias de Informação
VoIP	–	<i>Voice over Internet Protocol</i>
XML	–	<i>eXtensible Markup Language</i>

XMPP	–	<i>eXtensible Messaging and Presence Protocol</i>
XOL	–	<i>Ontology Exchange Language</i>
XSL	–	<i>Extensible Style Language</i>

Capítulo 1

1 Introdução

Neste primeiro capítulo, pretendo enquadrar o leitor no tema da presente Dissertação, no âmbito do Mestrado em Redes e Serviços de Comunicação. Nesse sentido, irei apresentar a área temática em que se insere esta dissertação, os problemas que se colocam a uma arquitectura de gestão onde aplicações tradicionais, Aplicações Autónomas e administradores de rede têm necessidade de comunicar, bem como os objectivos a que me propus ao desenvolver um trabalho, numa área, onde seguramente nos próximos anos as evoluções serão muitas e o todo o trabalho de investigação nesta área será de valor acrescentado muito importante.

1.1 Tema

Quando falamos em gestão de redes IP cada vez mais se verifica a necessidade de acções proactivas que, pelos seus próprios requisitos, levam a que a acção humana passe a dar lugar a acções autónomas com um certo grau de inteligência. Tais acções só são possíveis de implementar se falarmos em Aplicações Autónomas. Se pensarmos nestas aplicações, facilmente nos apercebemos que elas poderão ser utilizadas para gerir proactivamente uma rede IP a diferentes níveis (elementos, recursos, serviços, utilizadores, etc.) [1]. Pela sua filosofia, e do ponto de vista humano, a gestão fica mais facilitada.

As aplicações de gestão são usadas para implementar um modelo de informação de gestão comum. Geralmente os alarmes disparados pelas aplicações de gestão são enviados para uma consola de gestão, enviados por e-mail, ou outra forma de comunicação, para o gestor da rede. É certo e sabido que os gestores de rede nem sempre têm permanente contacto visual com o monitor, ou então, não estão presentes no local. Estas situações podem levar à criação de atrasos na resolução de problemas.

Para acelerar a resolução destes atrasos, as Aplicações Autónomas deverão ter aqui um papel importante, na medida em que podem elas mesmas tratar de resolver os problemas de rede, sem intervenção humana.

Um dos problemas que se antevê desde já é a comunicação *on-line* dos intervenientes de gestão, em resposta aos alarmes gerados pelos Sistemas de Gestão. Isto é, qual a forma que Utilizadores e Aplicações Autónomas têm para interagirem, ou negociarem entre si, acções de gestão, de forma a resolverem os problemas.

Assim, apesar de já existirem conceitos, como o FIPA e KQML na abordagem da comunicação entre agentes, estes não prevêem a interacção humana. Esta Tese, pretende explorar a utilização de *Instant Messaging* como forma de comunicação entre Aplicações Autónomas e Utilizadores na Gestão de Redes, tendo por base um modelo global de informação baseado no LDAP e utilizar o protocolo XMPP como protocolo base de comunicação entre aplicações e transporte de conteúdo XML.

1.2 Objectivos

No seguimento do exposto, o objectivo principal é incluir o *Instant Messaging* na Arquitectura de Gestão. Para alcançar este objectivo é proposta uma Arquitectura Multi-Agente, baseada em *Instant Messaging* como forma de comunicação de informação entre Utilizadores humanos e Aplicações Autónomas.

A Arquitectura de Gestão tem como base, um novo modelo de gestão de informação, baseado em Árvores de Informação e no LDAP como forma de lhe aceder, que se chama de LMIB. Este modelo foi desenvolvido para permitir a partilha de informação entre as várias aplicações de gestão de forma fácil, aberta e normalizada, sem recorrer a interfaces/APIs proprietárias.

A Arquitectura Multi-Agente, será composta por Aplicações Autónomas que agem em nome de um utilizador, ou em resultado de alarmes disparados na rede, por Utilizadores humanos que podem alterar o rumo das acções a tomar pelas Aplicações Autónomas, e finalmente pelo LDAP, responsável por implementar o modelo global de informação.

Sendo assim, é necessário que esta arquitectura tenha um meio de comunicação eficiente, isto é, possuir um mecanismo que possibilite a troca de mensagens *on-line*, entre Utilizadores humanos e Aplicações Autónomas.

Acredita-mos que o *Instant Messaging* pode desempenhar o papel de linguagem de conteúdo, através do uso do protocolo XMPP para transporte de conteúdo XML.

Sendo o mecanismo de transporte de mensagens uma linguagem de comunicação, torna-se necessária a compreensão do seu conteúdo. Aqui, entram em acção as chamadas linguagens de conteúdo que fornecem os aspectos sintácticos da representação da troca de informações e de conhecimento passados através de mensagens. Neste sentido, torna-se necessário especificar uma linguagem de conteúdo que permita às referidas Aplicações Autónomas e Utilizadores, o mútuo entendimento e conhecimento das tarefas respectivamente atribuídas na arquitectura.

1.3 Estrutura do Tese

Esta Tese está organizada em seis capítulos. Neste primeiro capítulo é apresentada uma introdução referindo o tema e objectivos, bem como uma descrição da organização deste relatório.

No segundo capítulo, é feito um estudo da tecnologia de agentes de software, dos sistemas existentes, das suas linguagens, e são apresentados alguns dos benefícios e expectativas, bem como limitações e desafios do uso deste tipo de tecnologia.

O terceiro capítulo, destina-se à descrição da Arquitectura de Gestão MANIA, onde se fala do Modelo de Gestão de Redes, do Modelo de Informação, mais concretamente o Sistema de Informação de Gestão de Redes, e do Barramento de informação de gestão de rede, que será a base do presente trabalho.

No quarto capítulo, é feita a descrição da Arquitectura de Comunicação do Sistema de Aplicações Autónomas de Gestão, onde é definido o Modelo de Informação, as Aplicações de Gestão, a Plataforma e Linguagens de Comunicação.

O quinto capítulo, contém o protótipo de utilização para ambientes de gestão Intra e Inter-Domínio.

Por último, no sexto capítulo são apresentadas as conclusões do trabalho descrito neste documento, faço também um balanço entre os objectivos propostos e o que foi feito e sugiro alguns melhoramentos.

Capítulo 2

2 Estado da Arte

Normalmente falamos em gestão de redes quando nos referimos a um mecanismo que age globalmente em muitas entidades e tende a fazer alterações na configuração de um sistema, levando-o de um estado estável para outro. As tarefas de gestão agem normalmente sob uma escala temporal relativamente grande.

O controlo de um sistema de telecomunicações pode, resumidamente, entender-se como acções muito pequenas necessárias para manter o mesmo em condições estáveis. No entanto, este conceito tende a tornar-se um pouco confuso com o aparecimento da nova tecnologia dos agentes, onde o conhecimento localizado dos próprios agentes permite a adaptação lógica, distribuída e coordenada, do controlo do equipamento de rede. Por exemplo, uma alteração muito pequena do funcionamento de um equipamento invocada de um ponto geograficamente distante. Para se poder aplicar esta tecnologia de agentes, equipas de investigação nesta área têm-se preocupado com a integração de conceitos de controlo e de gestão nestes sistemas, aplicando ainda conceitos económicos e conceitos aplicados na área da inteligência artificial na adjudicação de todos os recursos dos mesmos.

2.1 Definição de Agente Autónomo

Tem sido largamente reconhecido que o futuro das infra-estruturas das telecomunicações se tem tornado um campo bastante fértil para a utilização de aplicações baseadas em aplicações de software, [2] [3] afirmaram que "Os serviços das futuras redes de telecomunicações serão "povoados" por aplicações de software (agentes) capazes de assistirem um utilizador final na utilização da cada vez mais crescente informação disponível".

O termo agente tem vindo a ser adoptado genericamente para descrever o conceito de uma entidade, desenvolvida em software, isto é, uma aplicação, que automatize determinadas tarefas consideradas como cansativas e rotineiras para um humano. O aparecimento dos agentes foi apoiado no conceito básico da utilização de processos de software separados, implementados por exemplo, através da invocação de

procedimentos, processos leves – "*threads*" para tratarem de tarefas de controlo automatizado. Por este motivo, os agentes tornaram-se desde os anos noventa numa das maiores tecnologias de software. Um dos maiores domínios potenciais de aplicação para os agentes é o da indústria das telecomunicações, já que possibilita a gestão automatizada e o controlo da mesma.

Facilmente se percebe que os sistemas destas áreas são sistemas distribuídos e heterogéneos. Aqui não se pode falar simplesmente em agentes, mas sim em sistemas Multi-Agente – MAS, definidos como um conjunto de agentes que interagem uns com os outros e com o meio que os rodeia, com o propósito de, coordenadamente, resolver um problema particular. Tal pode ser realizado através da utilização de agentes individuais ou através de um sistema global cooperativo optimizado.

E notório que a aproximação efectuada pelos agentes tem influenciado significativamente a área das redes de telecomunicações, no sentido de se não pensar na informação como globalmente centralizada, mas sim em termos de sistemas abertos onde a distribuição geográfica e o controlo dessa informação aumentam a capacidade de processamento e a necessidade de fornecer serviços cada vez mais específicos.

Como foi referido, um agente pode ser descrito como uma aplicação de software, capaz de manipular de uma forma autónoma a selecção de acções quando ocorre determinado evento. Assim, os agentes possuem diferentes habilidades e qualidades, próprias para lidarem com o seu mundo ou domínio de aplicação. Tendo sido utilizado desde há bastante tempo, na área da computação distribuída, o termo agente tem sido aplicado na referência a entidades específicas – Cliente/Servidor – na resolução de problemas nos sistemas distribuídos de processamento.

Entre os atributos das actividades dos agentes podemos encontrar sociabilidade, autonomia, reactividade, adaptabilidade, pro-actividade, aprendizagem e grau de granularidade, descritos a seguir:

- **Sociabilidade** – um agente é capaz de utilizar a comunicação como base para a troca e partilha de informação com os restantes agentes do seu meio. Desta forma, os agentes operam à volta de um objectivo global;
- **Autonomia** – os agentes devem operar sem a intervenção de elementos externos, outros agentes ou humanos. Eles devem possuir algum tipo de controlo sobre as suas acções e sobre os seus estados internos;
- **Reactividade** – os agentes compreendem o seu ambiente e respondem atempadamente nas alterações que possam ocorrer;
- **Adaptabilidade** – os agentes são caracterizados pela sua flexibilidade, adaptação e facilidade para definir os seus próprios objectivos, baseando-se nos seus interesses. Uma das suas maiores características é a sua habilidade para adquirir e processar informação sobre determinada situação;
- **Pro-actividade** – os agentes devem exhibir publicamente o seu comportamento, isto é, mostrar que as acções tomadas produzem alterações benéficas ao seu ambiente. Esta capacidade requer muitas vezes a previsão de situações que possam vir a ocorrer, em vez de responder simplesmente a essas alterações;

- **Aprendizagem** – a comunidade de agentes pode, por si mesmo, possuir a habilidade de aprender, ou cada agente possuir um algoritmo próprio de aprendizagem. Esta aprendizagem permite muitas vezes a um agente alterar as suas futuras sequências de acção e de comportamento para que possíveis erros não ocorram. Aprender é muitas vezes um factor que fornece a um agente a habilidade de demonstrar um comportamento adaptativo.
- **Grau de Granularidade** – os agentes podem ter vários graus de complexidade. Os mais simples são caracterizados pela falta de inteligência, quando referidos sobre o seu comportamento. Estes agentes são denominados de reactivos. Os agentes mais complexos são denominados de cognitivos ou inteligentes e são caracterizados pela sua habilidade de conhecer o ambiente em que estão inseridos, agir sobre eles mesmos e no próprio ambiente. O comportamento observado pelos agentes é fruto da sua percepção, conhecimento e interacção.

Um dos maiores objectivos adjacentes na investigação desenvolvida na área da Inteligência Artificial – AI é o desenvolvimento de software com características semelhantes nas capacidades inteligentes dos seres humanos, tais como razão, comunicação através de uma linguagem natural, e aprendizagem. Além do mais, o crescente avanço das tecnologias na implementação das redes de computadores tem sido o grande dinamizador destes estudos, onde não se fala simplesmente em inteligência artificial, mas em Inteligência Artificial Distribuída – DAI.

2.2 Sistemas Multi-Agente – MAS

Aqui, a grande preocupação é o desenvolvimento de agentes onde exista interacção na resolução de problemas.

A um nível mais elevado, a aproximação aos sistemas Multi-Agente é bastante intuitiva e simples. Os investigadores baseiam-se na sua própria experiência para a resolução de problemas do mundo real, o que lhes dá a vantagem, de quando desenvolvem Aplicações Autónomas para a implementação de sistemas Multi-Agente, utilizarem uma metodologia natural e fiável na separação desses problemas em vários módulos. Por isso, o conceito de sistema Multi-Agente, permite adquirir uma conceitualização requerida no controlo descentralizado, como é o caso, por exemplo, da atribuição de papéis aos agentes. A aproximação usada no desenvolvimento de agentes oferece ainda uma analogia apropriada na decomposição de problemas e na delegação de tarefas, já que permite a abstracção de determinados problemas complexos que estão gradualmente a emergir do domínio das telecomunicações [4].

Segundo a literatura existente sobre a aproximação na resolução de problemas nos sistemas distribuídos com características acentuadamente heterogéneas, como é o caso da realidade das redes de telecomunicações, pode utilizar-se os sistemas Multi-Agente para, resumidamente:

- **Resolver problemas de dimensão elevada** para um simples agente centralizado devido, por exemplo a limitações de recursos ou impossibilidade de recuperação de falhas ou de acontecimentos inesperados;

- **Permitir a redução de custos de processamento** – é menos dispendioso em termos de hardware a utilização de processadores menos dispendiosos, do que a utilização de um único processador com equivalente capacidade de processamento;
- **Permitir a interligação e inter operação de sistemas distintos**, como é o caso de sistemas de apoio à decisão, dos sistemas de redes utilizando diferentes protocolos de comunicação, etc;
- **Melhorar a escalabilidade**, uma vez que a estrutura organizacional dos agentes pode ser alterada de uma forma dinâmica para exprimir o ambiente onde se encontram inseridos. Por exemplo, na medida que vai crescendo uma rede de comunicações, a organização dos agentes pode ser reestruturada pelos mesmos através da alteração dos seus papéis, crenças e acções;
- **Fornecer soluções:**
 - Inerentes aos problemas distribuídos;
 - Inferidas das fontes distribuídas de informação;
 - Onde o conhecimento é distribuído.

Adicionalmente, o paradigma dos sistemas Multi-Agente é inerentemente escalável devido à sua modularidade, uma vez que para se controlar sistemas complexos é necessário:

- Existência de coordenação entre os múltiplos objectivos, algumas vezes conflituosos, como é o caso de se ser responsável por objectivos com um nível superior de prioridade e continuar, em simultâneo, a prestar serviços a objectivos de prioridade inferior;
- Gerir dados adquiridos, muitas vezes incompletos e inconsistentes;
- Adoptar uma estratégia de controlo quando o ambiente, ou parte dele, sofre mutações drásticas, ou a estrutura de controlo deixa de funcionar normalmente.

Estes requisitos correspondem a muitos problemas encontrados na gestão de redes, como é o exemplo da comutação de pacotes de dados, implicando a ocorrência de um conjunto complexo de interacções. É extremamente difícil codificar cada possível cenário onde um grande número de incertezas pode potencialmente ocorrer em simultâneo.

Assim, requer-se uma técnica robusta para codificar todos esses factores que se interagem com o propósito da adaptação a alterações de estados nestes sistemas. A utilização de sistemas baseados em aplicações de agentes possui vantagens, face a sistemas de controlo centralizado, quando se pretende, por um lado, gerir muita informação e, por outro, atingir variados objectivos. Como consequência, as arquitecturas de gestão e controlo de redes de comunicações que utilizam a tecnologia de agentes são entendidas como um desafio e um passo importante a ser tomado no futuro dos sistemas deste tipo.

2.3 Coordenação

A resolução de um problema efectuada pelos agentes em sistemas MAS toma a forma de uma Resolução Distribuída do Problema – DPS e envolve coordenação, negociação e comunicação. De forma a serem resolvidos os problemas nestes sistemas, os agentes devem trocar a informação sobre as suas actividades. Além do mais, devem ainda coordenar essas actividades e negociar entre si quando aparecem determinados conflitos de forma a resolvê-los.

Estes conflitos podem variar desde simples limitações de contenção de recursos a questões mais complexas onde os agentes estão em desacordo devido a discrepâncias entre os seus domínios de conhecimento. Assim, é necessária uma coordenação para determinar a estrutura organizacional entre um grupo de agentes. Existem bastantes mecanismos que permitem esta coordenação, incluindo a estruturação dos sistemas de agentes em hierarquias. Desta forma, a coordenação torna-se o ponto central dos sistemas MAS.

Sem ela, desaparecem todos os benefícios existentes na interacção, e um grupo de agentes rapidamente degenera numa colecção de entidades com um comportamento caótico. A coordenação pode ser descrita essencialmente como um processo que garante a uma comunidade de agentes agir de uma forma coerente e harmoniosa. Como motivos para a existência de coordenação entre os agentes [5] encontram-se:

- Prevenção do caos, de uma maneira geral, nenhum agente possui uma visão global de toda a comunidade a que pertence, pois é simplesmente impraticável que tal aconteça numa comunidade razoavelmente complexa. Consequentemente, os agentes apenas possuem uma visão local dos objectivos e conhecimento que podem interferir com eles;
- Ir ao encontro de constrangimentos globais, os agentes que gerem uma rede devem responder atempadamente a certas falhas. Assim, é necessário um comportamento coordenado para que tal se consiga concretizar;
- Os agentes nos sistemas MAS possuem diversas capacidades e conhecimentos, muito análogo aos seres humanos, onde especialistas coordenam as suas capacidades, também os agentes devem coordenar as suas actividades;
- As acções dos agentes são frequentemente interdependentes, e por isso, um agente pode ter a necessidade de esperar que outro agente complete determinadas tarefas antes de executar as suas. Obviamente que estas tarefas mutuamente dependentes precisam de ser coordenadas.

2.4 Arquitecturas de Agentes

Outro aspecto não menos importante num agente é a sua arquitectura interna. A arquitectura interna de um agente está associada à própria definição e mecanismos de decisão do agente que determinam e influenciam a sua acção. Existem diversas abordagens ao nível dos mecanismos de decisão. Neste ponto existem várias abordagens conceptuais em relação à estrutura de um agente. Vamo-nos basear na classificação apresentada em [6] para as classificar. As arquitecturas dos agentes podem dividir-se em:

- Cognitivas;
- Reactivas;
- Híbridas.

2.4.1 Arquitecturas Cognitivas

A abordagem cognitiva ou deliberativa tem uma visão tradicional *top-down*, baseada no planeamento de estratégias que usam um modelo do mundo para verificar informação e gerar acções nesse mundo. Este tipo de abordagem permite analisar a qualidade do desempenho do agente. Normalmente, este tipo de agentes tem:

- Um objectivo;
- Conhecimentos e crenças acerca do seu contexto;
- Mecanismo de raciocínio.



Figura 1 – Arquitectura Cognitiva.

A formalização usada na representação interna dos agentes baseia-se normalmente em notações e teorias matemáticas com semânticas, lógicas modais que são estendidas para comportar o conceito de crença [6]. Uma crença pode ser vista como algo em que o agente acredita e utiliza nas suas decisões, mas que não é necessariamente verdade.

Este tipo de formalização permite uma abordagem simbólica e abstracta na análise do seu contexto, possibilitando a utilização de mecanismos de dedução e inferência, com as devidas adaptações. Por meio desta análise, o agente pode chegar a novas crenças acerca do seu contexto e agir sobre o mesmo.

Exemplos de utilização de agentes cognitivos podem ir desde agentes de planeamento (optimização e gestão de tarefas) a agentes autónomos que negociam na bolsa. Contudo este tipo de abordagem falha com o aumento da complexidade das situações, tendo um custo computacional "exponencial".

2.4.2 Arquitecturas Reactivas

A abordagem baseada em Arquitecturas Reactivas tem como base alguma reflexão feita em torno de animais mais simples, como formigas e abelhas. Apesar da simplicidade destes, o seu comportamento observado à luz da Etologia, ciência que estuda os comportamentos, revelava-se “inteligente” [7].

Os modelos reactivos encontram-se no lado oposto dos agentes cognitivos. O seu mecanismo de "inferência" é simplificado ao máximo e especializado na resposta a determinados "estímulos" do contexto. Rodney Brooks [7], um defensor desta abordagem sintetiza o apoio à abordagem reactiva:

- Melhor modelo do mundo é o próprio mundo;
- Comportamentos inteligentes emergem da dinâmica das interações do agente com o contexto.

Assim, a ênfase é colocada não na definição de modelos abstractos do mundo, mas em definir capacidades simples e úteis para os agentes. A arquitectura baseia-se na existência de pares estímulo/tarefa como mecanismos simples de reacção dos agentes. A base desta visão é que se espera que pela conjugação destas reacções e interações com o exterior possam nascer "abordagens" inteligentes por parte dos agentes. No âmbito deste tipo de arquitecturas surgiu uma corrente que introduz a noção de comportamento. A noção de comportamento pode ser vista como mecanismo de controlo que permite alcançar ou manter um dado objectivo ao nível do agente. Os comportamentos podem envolver a conjugação de várias “reacções” dos agentes na sua definição.

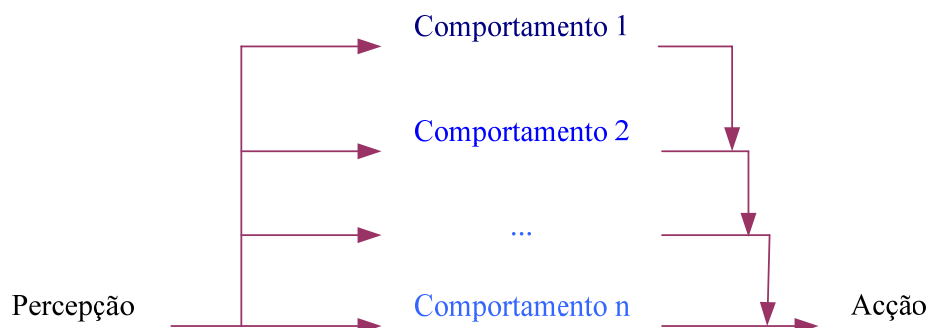


Figura 2 – Arquitectura de Subsunção.

Nesta linha lógica, Brooks sugere uma arquitectura, a arquitectura de subsunção, como base para a organização funcional dos comportamentos. A ideia base, é que inerente a dado agente reactivo, está associado um esquema de combinação dos comportamentos que permite descrever prioridades, acções conjuntas, etc. Esta abstracção permite de alguma forma organizar a estrutura funcional do agente. Este tipo de abordagem é viável como mostra o facto de terem sido escolhidos para os *Mars Rovers*, exploradores autónomos a usar na exploração de Marte, uma estrutura "inteligente" deste tipo. A descrição das regras é muitas vezes em instruções de baixo nível que são usadas como rotinas para descrever e reagir aos estímulos, embora aquelas possam ser representadas em lógica, linguagens predefinidas, etc.

No extremo das abordagens reactivas estão as abordagens puramente reactivas, conhecidas também por reflexivas, onde a ideia base consiste num agente ter um conjunto de acções pré programadas com as quais reage a estímulos bem definidos, sem qualquer mecanismo de "análise", à semelhança dos reflexos humanos. Este tipo de sistemas, apesar de se terem mostrado altamente adaptáveis, são ao mesmo tempo, pouco flexíveis, no sentido de perseguirem novos objectivos, pois não são capazes de memorizar informação dinamicamente.

2.4.3 Arquitecturas Híbridas

Os modelos híbridos encontram-se estruturados em níveis [6]:

- Níveis essencialmente reactivos;
- Níveis mais cognitivos (planeamento e "inferência" sobre os conhecimentos de contexto).

A ideia base, é que as tarefas que asseguram a sobrevivência do agente, tenham uma estrutura mais reactiva e todo o mecanismo cognitivo está encarregue de delinear as estratégias para alcançar os objectivos do agente.

2.5 Modelo Tradicional de Agentes FIPA

A arquitectura Multi-Agente tradicional, definida pela FIPA, define o modelo de referência lógica para a criação, registo, localização e comunicação dos agentes [8].

Este modelo é composto por vários blocos conforme se verifica na Figura 3. De notar que as entidades presentes neste modelo de referência são um conjunto de funcionalidades lógicas e não reportam nenhuma configuração física.

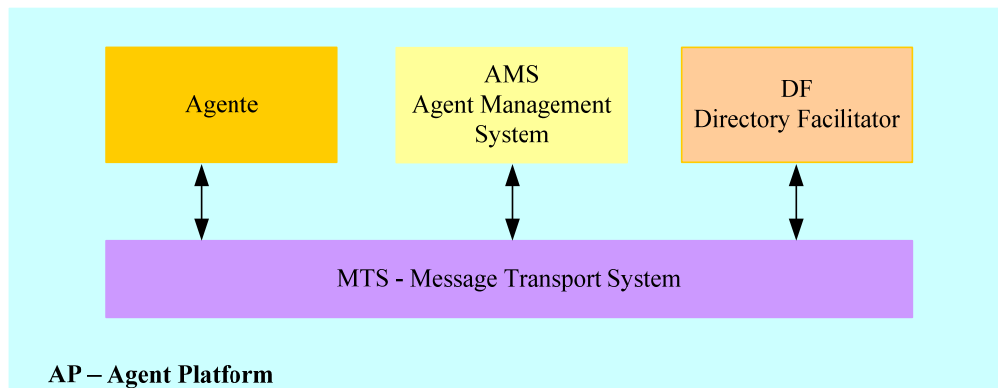


Figura 3 – Modelo de referência para Agentes FIPA.

2.5.1 Agente

Um Agente é um processo computacional, que implementa a funcionalidade de comunicação de uma aplicação.

O Agente é o elemento principal numa plataforma de agentes, na medida em que combina o(s) serviço(s) que implementa, num modelo unificado e integrado de execução. Um Agente tem de ter uma identidade, que permite distinguir os agentes uns

dos outros dentro do seu Universo e comunicam através do uso de Linguagem de Comunicação de Agentes – ACL.

2.5.2 Facilitador

O Facilitador é o elemento que fornece serviço de páginas amarelas a outros agentes, apesar de ser um componente opcional da plataforma de agentes.

É responsável por disponibilizar informação actual sobre os agentes que estão no seu directório, e a todos os agentes autorizados, numa base não discriminatória. Os Agentes podem registar os seus serviços com o Facilitador, ou então questionar este, para descobrir quais os serviços oferecidos por outros agentes.

Uma Plataforma de Agentes pode suportar vários Facilitadores, e estes podem-se registar uns nos outros de forma a formarem federações.

Qualquer agente que queira publicar os seus serviços a outros agentes, deve ser capaz de encontrar um Facilitador apropriado e pedir para registar a sua descrição. Um agente pode recusar um pedido para um serviço que é anunciado através do Facilitador. Adicionalmente, o Facilitador não pode garantir a validade ou precisão da informação registada, nem pode controlar o ciclo de vida dos agentes.

Deve ser fornecida uma descrição do objecto que contém valores para todos os parâmetros obrigatórios da descrição. Também podem ser fornecidos parâmetros opcionais e privados, contendo informação não normalizada, que um programador queira incluir no directório.

A qualquer altura ou por qualquer motivo, um agente pode pedir ao Facilitador para modificar a sua descrição. Um agente também pode pesquisar informação num Facilitador.

Por outro lado, o Facilitador não garante nenhuma validade da informação fornecida na resposta à pesquisa, uma vez que este não impõe nenhuma restrição da informação que pode ser registada no mesmo. Contudo, o Facilitador pode restringir o acesso à informação no seu directório e verificar as permissões para os agentes.

2.5.2.1 Funções suportadas pelo Facilitador

De forma a aceder ao directório de descrições dos agentes geridos pelo Facilitador, cada Facilitador deve ser capaz de executar as seguintes funções.

- register;
- deregister;
- modify;
- search.

2.5.2.2 Facilitadores Federados

Conforme dito anteriormente, uma Plataforma de Agentes pode suportar vários Facilitadores, e estes podem-se registar uns nos outros de forma a formarem federações. Esta situação tem particular interesse no que diz respeito a pesquisas.

O Facilitador tem um mecanismo de pesquisa, que primeiro pesquisa localmente e depois estende a pesquisa a outros Facilitadores, se permitido para tal. O mecanismo de pesquisa por defeito é assumido como uma pesquisa baseada em `depth-first` em todos os Facilitadores. A federação de Facilitadores pode ser usada para estender as pesquisas, através do registo de Facilitadores uns com os outros, e com o valor `fipa-df` no parâmetro `type` na descrição do serviço.

Quando um Facilitador recebe uma acção de pesquisa, ele pode determinar se precisa de propagar a pesquisa a outros Facilitadores que estão registados com ele. As pesquisas só devem ser encaminhadas a outros Facilitadores, quando o valor do parâmetro `max-depth` for superior a 1, e não ter recebido uma pesquisa anterior com o mesmo parâmetro `search-id`.

Se a pesquisa for encaminhada, então devem ser cumpridas as seguintes regras:

- Não deve mudar o valor do parâmetro `search-id`, quando propaga a pesquisa e o valor de todos os parâmetros `search-id` devem ser globalmente únicos;
- Antes da propagação, deve ser decrementado de um o valor do parâmetro `max-depth`.

2.5.3 Sistema de Gestão de Agentes

O Sistema de Gestão de Agentes é responsável por gerir a operação de uma Plataforma de Agentes, na criação e eliminação de agentes, sendo por este motivo uma componente obrigatória de uma plataforma de agentes.

Um Sistema de Gestão de Agentes pode pedir aos agentes para executarem funções específicas como desistirem, isto é, terminar toda a execução na sua Plataforma de Agentes, bem como a possibilidade de forçar uma função, se a mesma tivesse sido ignorada pelo agente em causa.

Por cada Plataforma de Agentes só pode existir um Sistema de Gestão de Agentes.

Este componente mantém um directório de identificadores, que contém os endereços de transporte, entre outras coisas, para os agentes registados com a plataforma de agentes.

Todos os índices dos agentes que residem na Plataforma de Agentes são mantidos pelo Sistema de Gestão de Agentes, o que inclui os Identificadores dos mesmos.

A residência de um agente, numa Plataforma de Agentes implica que o agente se tenha registado com o Sistema de Gestão de Agentes. Cada agente, de forma a cumprir o modelo de referência FIPA, tem de se registar com o Sistema de Gestão de Agentes da sua Plataforma de Agentes de origem ou HAP – *Home Agent Platform*.

Cada Agente tem de se registar com um Sistema de Gestão de Agentes, de forma a obter um identificador válido. Pode-se dizer que o Sistema de Gestão de Agentes oferece o serviço de páginas brancas a outros agentes.

As descrições dos agentes podem ser modificadas a qualquer altura e por qualquer razão. O processo de modificação é restrito pela autorização do Sistema de Gestão de

Agentes. A vida de um agente numa Plataforma de Agentes termina aquando do seu desregisto do Sistema de Gestão de Agentes.

Após o desregisto, o Identificador do agente não pode ser removido do directório e pode ser libertado a outros agentes. A descrição dos agentes pode ser pesquisada no Sistema de Gestão de Agentes, e o acesso ao directório `ams-agent-descriptions` é controlado por este. O Sistema de Gestão de Agentes também contém a descrição da Plataforma de Agentes, que pode ser obtida através do pedido de execução da função `get-description`.

2.5.3.1 Funções suportadas pelo Sistema de Gestão de Agentes

- `register`;
- `deregister`;
- `modify`;
- `search`;
- `get-description`.

2.5.3.2 Funções suportadas pelo Sistema de Gestão de Agentes sobre a Plataforma de Agentes

- `suspend-agent`;
- `terminate-agent`;
- `create-agent`;
- `resume-agent-execution`;
- `invoke-agent`;
- `execute-agent`;
- `resource-management`.

2.5.4 Serviço de Transporte de Mensagens

O Serviço de Transporte de Mensagens é o método de comunicação usado entre agentes, em plataformas de agentes diferentes.

Este serviço é responsável pela entrega de mensagens entre agentes residentes na mesma Plataforma de Agentes e para agentes que residem noutras Plataformas. Todos os agentes FIPA têm, acesso a pelo menos um Serviço de Transporte de Mensagens, e só as mensagens endereçadas para um agente é que podem ser enviadas pelo Serviço de Transporte de Mensagens.

O conteúdo das mensagens é representado através do uso de ACLs, que por sua vez são transportadas através do Serviço de Transporte de Mensagens e de Protocolos de Transporte de Mensagens.

2.5.4.1 Protocolo IIOP

Este Serviço de Transporte de Mensagens é baseado na transferência de uma estrutura OMG IDL [9], que contém o envelope da mensagem e uma sequência de octetos que

representam o corpo da mensagem ACL. O envelope e o corpo de mensagem são transferidos juntos através da invocação IOP [OMGiop] de sentido único.

Assim que o pedido for recebido, o envelope da mensagem é usado pelo ACC para obter as instruções e a informação necessária para tratar correctamente o corpo de mensagem.

2.5.4.2 Protocolo HTTP

Neste caso o Serviço de Transporte de Mensagens é baseado na transferência dos dados que representam a mensagem do agente, incluindo o envelope da mensagem num pedido HTTP [10]. A transferência de dados HTTP é um processo com dois passos:

- O emissor faz um pedido HTTP e após ter recebido os dados, o receptor emite uma resposta HTTP;
- O receptor faz o *parsing* do envelope da mensagem e a mensagem é tratada de acordo com as instruções e informação recebida no envelope da mensagem.

2.5.5 Plataforma de Agentes

Uma Plataforma de Agentes é uma infra-estrutura física sobre a qual os agentes podem ser programados e desenvolvidos. A Plataforma de Agentes consiste em máquina(s), sistema operativo, software de suporte para agentes, componentes de gestão de agentes FIPA (DF, AMS e MTS) e agentes.

Os agentes podem trocar mensagens directamente uns com os outros, por qualquer método que eles possam suportar.

2.5.6 Identificação dos Agentes

O modelo de referência de nomes dos agentes FIPA, identifica um agente como uma colecção extensível de pares parâmetro-valor chamada de *Agent Identifier* – AID. A natureza extensível do identificador permite acolher outros requisitos, como nomes sociais, alcunhas, roles ou papéis, que podem ser anexados a serviços na Plataforma de Agentes.

Um *Agent Identifier* é composto por:

- Parâmetro *name*, é o identificador global único que pode ser usado como uma referência única do agente. Um dos mecanismos para construir o parâmetro *name* é a partir do nome actual do agente e do endereço da plataforma à qual o agente pertence, separado pelo carácter @;
- Parâmetro *address*, é uma lista de endereços de transporte onde as mensagens podem ser entregues;
- Parâmetro *resolvers*, contém uma sequência de *Agent Identifiers*, a partir da qual o identificador do Agente pode ser resolvido num endereço de transporte ou conjunto de endereços de transporte;

- Dois *Agent Identifiers* são considerados equivalentes se o parâmetro *name* for o mesmo.

2.5.6.1 Endereço de Transporte

Um endereço de transporte é um endereço físico a partir do qual um agente pode ser contactado. Geralmente este endereço é específico ao protocolo de transporte de mensagens. Um agente pode suportar muitos métodos de comunicação e podem colocar múltiplos endereços de transporte no parâmetro *Agent Identifier*.

2.5.6.2 Resolução de Nomes

A resolução de nomes é um serviço que é fornecido pelo Sistema de Gestão de Agentes, através da função de pesquisa (*search*). O parâmetro *resolvers* do *Agent Identifier* contém uma sequência de Identificadores a partir da qual o *Agent Identifier* do agente pode ser resolvido num endereço de transporte.

2.6 Linguagem de Comunicação de Agentes

As linguagens de comunicação possibilitam aos agentes, a troca de informações e de conhecimento [11]. No entanto, antes do seu aparecimento, usavam-se outros métodos para mostrar uma semelhança com a troca de informação e de conhecimento entre aplicações, desde a evocação remota de procedimentos, RPC e RMI, até ao CORBA, os objectivos eram os mesmos.

O que distingue as linguagens de comunicação desses esforços passados são os objectos do discurso e a complexidade da sua semântica. As linguagens de comunicação para agentes encontram-se a um nível superior, quando comparados com esses métodos, devido a duas grandes razões:

1. Porque tratam proposições, regras e acções em vez de simples objectos sem semântica associada;
2. Numa linguagem de comunicação uma mensagem descreve um estado desejado através de uma linguagem declarativa em vez de um método ou função.

Contudo, estas linguagens não cobrem toda a gama de aplicações onde poderá ocorrer troca de informações. Os agentes podem, e devem, trocar objectos mais complexos tais como objectivos e expectativas, ou mesmo partilhar experiências e estratégias de actuação a longo prazo. A um nível mais técnico, quando se utiliza uma linguagem de comunicação, os agentes trocam mensagens através da rede utilizando um protocolo de baixo nível como, por exemplo, o SMTP, o TCP/IP ou o HTTP.

A linguagem de comunicação define, por si só, o tipo de mensagens, e seu significado, que os agentes podem trocar entre si. No entanto, eles não podem ocupar-se apenas de mensagens simples. Eles possuem tarefas orientadas a conversações, isto é, sequências de mensagens partilhadas que seguem, tais como negociações e actuações.

De uma forma geral, os diferentes tipos de mensagens das linguagens de comunicação são entendidos como sendo actos de comunicação que, por seu turno, são

usualmente utilizados para descrever e definir crenças, desejos e intenções ou BDI – *Beliefs, Desires and Intentions* dos próprios agentes. Este tipo de descrição ao nível intencional pode tornar-se numa forma útil de se ver um sistema, possuindo um aspecto concreto de processamento computacional.

Neste caso, pode desenvolver-se uma larga gama de agentes BDI, os quais podem possuir representações implícitas e explícitas dessas correspondentes modalidades (crenças, desejos e intenções). Essas representações são construídas através de uma descrição conceptual do conhecimento de um agente, e dos seus objectivos e compromissos, também conhecida como teoria BDI [12].

2.6.1 A Linguagem KQML

A comunicação utilizando a linguagem KQML, envolve a consideração de três níveis, conteúdo, mensagem e comunicação. Cada um destes níveis corresponde a um conjunto de atributos associados à mensagem. Os atributos que efectivamente são mencionados numa mensagem dependem do tipo de mensagem em causa, podendo alguns ser opcionais. Nesta secção a caracterização das mensagens vai incluir apenas a definição dos seguintes atributos de cada um dos níveis:

- **Nível da mensagem** – Apenas a identificação da performativa utilizada, que identifica o tipo da mensagem;
- **Nível da comunicação** – Indicação, para os atributos *Sender* (emissor) e *Receiver* (receptor), do nível do agente correspondente, e para os atributos *Reply-With* e *In-Reply-To*, da identificação da mensagem em causa;
- **Nível de conteúdo** – Uma expressão que caracteriza o conteúdo da mensagem, atributo *Content*, mas apenas a um nível conceptual.

No entanto, só serão detalhadas as mensagens que têm conteúdo. Cada mensagem será identificada da seguinte maneira:

Tabela 1 – Identificação de Mensagens KQML.

Performativa:	<Nome da performativa>
<i>Sender:</i>	<Identificação do agente emissor>
<i>Receiver:</i>	<Identificação do agente receptor>
<i>In-Reply-To:</i>	<Identificador da mensagem à qual se está a responder>
<i>Reply-With:</i>	<Identificador da mensagem para referência futura>
<i>Content:</i>	<Expressão com o conteúdo da mensagem>

Nem todos os atributos aparecem em todas as mensagens. Em KQML podem existir mensagens sem conteúdo, com o atributo *Content* vazio. Por exemplo, uma mensagem do tipo *discard* (que informa o receptor de que o emissor não está interessado em mais respostas a um pedido feito utilizando uma performativa multi-resposta) não tem

conteúdo, leva apenas a indicação (no atributo `In-Reply-To`) da mensagem à qual está a responder.

O conteúdo de uma mensagem pode ser outra mensagem. Em relação às mensagens sem conteúdo será apenas referido o contexto em que são usadas.

Em KQML é também possível especificar qual a linguagem em que está expresso o conteúdo das mensagens.

Na Tabela 2 é possível ver algumas performativas que foram definidas na linguagem KQML e uma breve descrição do significado de cada um delas.

Tabela 2 – Performativas KQML.

Categoria	Performativa	Descrição
Discursiva	Ask-if	O agente emissor pretende saber se o conteúdo da mensagem existe na base virtual de conhecimento do receptor.
	Ask-all	O agente emissor pretende conhecer todas as instâncias do conteúdo da mensagem existentes na base virtual de conhecimento do receptor.
	Ask-one	O agente emissor pretende conhecer uma instância do conteúdo da mensagem existente na base virtual de conhecimento do receptor.
	Stream-all	Resposta múltipla semelhante a ask-all.
	Eos	Finalização do pedido de resposta múltipla.
	Tell	Indicação ao agente receptor que o conteúdo da mensagem existe na base virtual de conhecimento do agente emissor.
	Untell	O conteúdo da mensagem não existe na base virtual de conhecimento de um agente.
	Deny	A negação do conteúdo existe na base virtual de conhecimento de um agente.
	Achieve	O agente emissor pretende que o agente receptor concretize uma acção correspondente com a veracidade do conteúdo da mensagem.
	Unachieve	Acto de correspondência inversa ao do achieve.
	Advertise	O agente emissor pretende que o agente receptor saiba que ele consegue processar o conteúdo da mensagem.

	Unadvertise	Cancelamento de um acto achieve anterior.
	Subscribe	O agente emissor pretende receber do agente receptor actualizações do conteúdo da mensagem.
Intervenção e Mecanismo	Error	O agente emissor considera que o conteúdo da mensagem anterior, recebida do agente receptor desta, se encontra mal indicado.
	Sorry	O agente emissor entende o conteúdo da mensagem anterior mas não o consegue processar.
	Standby	O agente emissor indica ao receptor que se encontra num estado de espera e pronto para processar o conteúdo da mensagem.
	ready	O agente emissor indica ao receptor que se encontra pronto para responder a uma mensagem prévia.
Facilitação e de Rede	Transport-address	O agente emissor associa o seu nome simbólico a um novo protocolo de transporte de mensagens.
	Forward	O agente emissor pretende fazer seguir uma mensagem recebida até ao agente receptor.
	Boker-one	O agente emissor pretende descobrir uma resposta a uma performativa, que pode ser dada por um outro agente.
	Broker-all	O agente emissor pretende descobrir todas as respostas a uma performativa, que podem ser dadas por outros agentes.
	Recruit-one	O agente emissor pretende que lhe seja indicado um agente capaz de responder a uma performativa.
	Recruit-all	O agente emissor pretende que lhe seja indicado todos os agentes capazes de responder a uma performativa.

Na Tabela 3 é possível ver um exemplo, de uma mensagem em que um agente A faz uma interrogação a outro agente sobre o preço das acções da Galp.

Nesta mensagem a performativa é ask-one, o conteúdo (:content (PRICE Galp ?price)), e a ontologia assumida é identificada pela expressão :ontology Psi20.

Tabela 3 – Exemplo de mensagem KQML.

Exemplo de mensagem KQML

```
(ask-one
  :sender A
  :receiver stock-server
  :reply-with Galp-stock
  :language LPROLOG
  :ontology Psi20
  :content (PRICE Galp ?price))
```

2.6.1.1 Vantagens e limitações da linguagem KQML.

A principal vantagem da linguagem KQML é a sua capacidade para suportar várias arquitecturas diferentes de agentes com seu conjunto extensível de Performativas. Como resultado, o KQML tornou-se uma linguagem padrão para comunicação entre agentes em diferentes áreas de aplicação.

No entanto, a sua primeira versão teve algumas críticas como pode ser observado em [13], que apontam para uma confusão no uso destas Performativas. De certa forma, devido a esta fraqueza, vários dialectos de KQML surgiram na indústria e foram usados em diferentes projectos de agentes de software que não podem inter operar.

2.6.2 A linguagem FIPA-ACL

A especificação FIPA-ACL [14] é baseada na troca de mensagens que os agentes utilizam para comunicar. Ela define uma linguagem normalizada de mensagens, especificando a codificação, a semântica e os factos a elas ligados, não sendo, no entanto, definido especificamente o mecanismo para o transporte interno das mensagens, já que os agentes podem situar-se em plataformas diferentes e utilizarem diferentes tecnologias de rede.

Neste sentido, o grupo FIPA especificou que as mensagens transportadas entre plataformas devem ser codificadas numa forma textual. Assim, assume-se que um agente deva possuir algum meio de transmissão dessa forma textual.

Na tabela seguinte é possível ver os parâmetros predefinidos pelo FIPA-ACL.

Tabela 4 – Parâmetros predefinidos no FIPA-ACL.

Palavra Reservada	Descrição
:sender	Agente emissor do acto de comunicação.
:receiver	Agente receptor do acto de comunicação.
:content	O próprio conteúdo da mensagem pretendida ser trocada.
:reply-with	Identificação esperada para uma resposta à mensagem actual.
:in-reply-to	Identificação esperada para uma resposta à mensagem anterior.

:envelope	Informação adicional útil enviada pelo serviço de transporte, tal como a hora de envio e recepção da mensagem.
:language	Identifica a linguagem de conteúdo utilizada.
:ontology	Indica em que termos (ontologias) o conteúdo da mensagem possui algum significado.
:reply-by	Indica a que horas ou data o agente emissor deseja receber uma resposta.
:protocol	Introduz um identificador que mostre o protocolo utilizado pelo agente emissor.
:conversation-id	Expressão utilizada para identificar uma sequência de actos de comunicação em trânsito que, no seu todo, constituem uma conversação.

A linguagem de comunicação para agentes FIPA-ACL é muito similar ao KQML, sendo a sua sintaxe idêntica, exceptuando-se a nomenclatura de algumas performativas primitivas.

Desta forma, é mantida a aproximação utilizada pelo KQML no que se refere à separação da linguagem exterior, que define o significado da intenção da mensagem, da linguagem interior ou de conteúdo que indica a expressão aplicável ao que o interlocutor acredita, deseja ou tenciona.

Tabela 5 – Performativas FIPA-ACL.

Performativas	Descrição
accept-proposal	Acto de aceitar uma proposta submetida anteriormente.
agree	Acto de concordar numa possível forma de actuar.
cancel	Acto de cancelar alguma acção pedida anteriormente.
cfp	Acto de pedir propostas para se efectuar determinada acção.
Confirm	O emissor informa o receptor que determinada proposição é verdadeira.
Disconfirm	O emissor informa o receptor que determinada proposição é falsa.
Failure	Acto de informar outro agente que foi tentado efectuar uma acção, porém sem êxito.
Inform	O emissor informa o receptor que uma dada proposição é verdadeira.
inform-if	Acção não atómica para o agente da acção, correspondente a

	informar se o conteúdo é ou não uma proposição verdadeira.
inform-ref	Acção não atómica para o emissor no sentido de informar o receptor que objecto corresponde a um descriptor (por exemplo, um nome).
not-understood	O emissor do acto informa o receptor que não entendeu a acção pretendida.
Propose	Acto de submeter uma proposta para concretizar determinada acção.
query-if	Acto de perguntar a outro agente se uma proposição é ou não verdadeira.
query-ref	Acto de perguntar a outro agente por um objecto referente a uma expressão.
Refuse	Acto de recusar efectuar uma acção e explicar porquê.
reject-proposal	Acto de rejeitar uma proposta durante uma negociação.
request	O emissor pede ao receptor para efectuar uma acção.
request-when	O emissor pede ao receptor para efectuar uma acção quando uma dada proposição for verdadeira.
request-when-ever	O emissor pede ao receptor para efectuar uma acção, assim e sempre que uma dada proposição for verdadeira.
subscribe	Acto de requerer uma actualização constante de um objecto identificado.

Na tabela seguinte, é apresentado um cenário de comunicação onde um agente B pede a uma agente C para lhe indicar o actual Presidente de Portugal.

Tabela 6 – Exemplo de mensagem FIPA-ACL.

Exemplo de mensagem FIPA-ACL

```
(request
  :sender I
  :receiver C
    :content (inform-ref
      :sender C
      :receiver B
      :content (iota ?x (PTPresident ?x))
      :ontology world-politics
      :language sl
```

```
)  
:reply-with query0  
:language sl)
```

2.6.2.1 Vantagens e limitações da linguagem FIPA-ACL

A FIPA-ACL é uma linguagem de comunicação para agentes que envolve muitas das particularidades da indústria e da academia. A FIPA-ACL baseia-se em componentes práticos de comunicação e cooperação inter agente com uma semântica formal definida.

Estas características tornam a FIPA-ACL um ponto de referência.

2.7 Os Agentes na Gestão de Redes

Os agentes podem ser vistos como uma tecnologia tridimensional [1], uma vez que exibem propriedades características, comunicam por intermédio de um conjunto de linguagens e são desenvolvidos seguindo determinadas arquitecturas.

Estas três dimensões tornam os agentes especialmente apropriados na substituição do Homem por aplicações de software baseados na tecnologia dos agentes. É muito comum, quando nos referimos na gestão e controlo de redes, encontrar operadores humanos a interagirem com utilizadores finais. As quatro propriedades mais importantes dos agentes, autonomia, actividade, reactividade e sociabilidade, são características fundamentais para aplicações que funcionam em nome de um utilizador final ou de uma operadora fornecedora de serviços, de forma a negociar contratos de tráfego, ou encontrar e filtrar informações de gestão num ambiente distribuído e complexo. Para que os agentes possam interagir de forma eficiente é imperativa a existência de três componentes fundamentais:

- Uma linguagem comum;
- Uma compreensão comum do conhecimento trocado;
- Habilidade para trocar tudo o que é definido no contexto de actuação [15].

No entanto, os agentes para comunicarem, utilizam diferentes tipos de linguagens de comunicação que lhes permitem trocar informações correspondentes com as suas propriedades e com as tarefas que enfrentam. Os agentes utilizam estas linguagens de comunicação para desenvolverem actos de comunicação do tipo declarativo, interrogativo e permissivo. A semântica da informação passada através dos actos de comunicação e expressa por linguagens de conteúdo que podem ser estendidas a ontologias particulares, associadas ao conteúdo. Uma arquitectura Multi-Agente deve basear-se numa de três aproximações, reactiva, deliberativa [16] ou híbrida [17], e adaptar-se ao domínio do problema, especialmente no que diz respeito na parte reactiva, a responsável pelo ponto de ligação dos agentes ao meio real.

Que tipo de comunicação deve um agente ter com outro? (devemos minimizar a comunicação entre eles para que uma acção decidida seja realizada mais rapidamente);

- Onde devem estar localizados os agentes? Um agente por ponto de rede? Um agente em cada ligação física? Um a representar cada ligação?
- Devem os agentes ser estáticos ou móveis?
- Que visão e conhecimento de uma rede deve possuir um agente? É benéfico que um agente tenha o conhecimento de eventos indirectamente com ele relacionados e que ocorrem localmente a outro agente?
- Que grau de dependência deve existir entre os agentes nos sistemas? Se um agente morre continuará o sistema robusto?

2.8 Benefícios e Expectativas dos Sistemas de Agentes

Os principais benefícios e expectativas na adopção de agentes são, entre outros, os seguintes:

Delegação: A delegação corresponde ao modelo de interacção homem-máquina de gestão indirecta. O utilizador em vez de executar directamente um conjunto de acções, apenas “comanda ou programa” um determinado agente especializado que lhe realiza as tarefas pretendidas de forma assíncrona e eventualmente desconectada.

Personalização: A personalização de determinados serviços permite que estes se adaptem às exigências e preferências de cada utilizador e pode constituir um factor competitivo decisivo em determinadas áreas de negócio. Nomeadamente, na área do comércio electrónico, da banca, ou da disseminação electrónica de informação.

Automatização: A capacidade de automatização providenciada por um agente é particularmente adequada na realização de tarefas rotineiras, sistemáticas e eventualmente complexas. As áreas de aplicação variam significativamente. Por exemplo, agentes de interface que detectam padrões de sequências de acções realizadas repetidamente e automatizam-nas convenientemente, ou agentes responsáveis pela integração de informação nos vários sistemas internos e legados das organizações.

Notificação: A notificação de eventos por entidades de software evita que a mesma actividade tenha de ser realizada directamente por um utilizador. Através deste mecanismo, na ocorrência de determinado evento, o agente pode realizar um conjunto restrito de acções e/ou notificar o sucedido ao seu utilizador.

Resolução de tarefas complexas: Os sistemas de agentes são adequados à resolução de problemas particularmente complexos, ou cuja solução exija a participação de múltiplos intervenientes, já que a complexidade do sistema pode ser dividida por múltiplas entidades relativamente autónomas de complexidade inferior.

Mobilidade: A capacidade de mobilidade de agentes de software pode vir a potenciar novas aplicações e melhorar o desempenho e escalabilidade de aplicações distribuídas em contextos abertos, como a Internet e em ambientes de computação móvel. Contudo, e por outro lado, os agentes móveis levantam alguns desafios, como sejam a segurança e a robustez.

Aprendizagem: A capacidade de aprendizagem de um agente pode contribuir significativamente para um aumento da qualidade geral de um serviço. Por exemplo, através do reconhecimento de padrões de compra de um cliente, de forma a refinar continua e implicitamente o seu perfil, ou através do refinamento dinâmico do perfil de utilizador, de forma a melhorar a eficácia de um mecanismo de filtragem de informação.

2.9 Limitações e Desafios dos Sistemas de Agentes

Os principais desafios e limitações que se reconhecem na aplicação/utilização dos agentes no contexto da Internet, em particular do futuro da Internet, vista como um espaço aberto de agentes são entre outros:

Designação ambígua: O facto de um agente suscitar inúmeras interpretações, apresentar um conjunto de características alargado, ser influenciado por diferentes comunidades científicas e existir em diferentes áreas de aplicação, conduz a uma ambiguidade e confusão generalizada do que é, de facto, um agente de software. Tal ambiguidade pode conduzir facilmente a discussões pouco produtivas e a um consequente descrédito geral.

(des) Confiança do utilizador: O utilizador não está habituado, em geral, a interactivar de forma indirecta com o computador. Não está habituado a utilizar aplicações que lhe executam tarefas de forma autónoma. Entre outros, surge o “síndrome da competência e da confiança” referido por [18]. Por um lado, o utilizador tem de ter confiança nos agentes que utiliza, nomeadamente, confiança nos fabricantes dos agentes em causa, de forma a delegar-lhe tarefas relativamente complexas e/ou críticas. De forma a facilitar essa confiança, o utilizador deverá poder monitorizar o estado corrente dos seus agentes e eventualmente obter um registo de actividades realizadas, com maior ou menor detalhe. Por outro lado, os agentes têm de mostrar que são competentes/capazes de realizar adequadamente e com flexibilidade as tarefas para os quais foram desenvolvidos e configurados.

Novos modelos de negócio: Por vezes os problemas da adopção e utilização de agentes não têm a ver com questões tecnológicas, mas sim com a adequação e integração das novas aplicações aos modelos de negócio existentes ou emergentes. Ou seja, embora de um ponto de vista tecnológico seja possível, por exemplo conceber agentes de pesquisa do melhor preço de um produto, com eventual compra correspondente, ou agentes de jornais electrónicos personalizados, é fundamental que as soluções tecnológicas sejam acompanhadas por correspondentes soluções legais e de carácter ético.

(in) Experiência de desenvolvimento: A área de concepção e implementação de aplicações baseadas em agentes é relativamente recente, nomeadamente segundo a perspectiva da engenharia de software. Não existem soluções maduras de métodos de desenvolvimento de software para este conjunto de aplicações, nem tão pouco ambientes e ferramentas de suporte e de desenvolvimento.

(in) Segurança: Em particular no âmbito de aplicações baseadas em agentes móveis [19] [20] colocam-se questões importantes do ponto de vista da segurança. O facto de

um agente poder mover-se para uma máquina remota, e aí executar-se, exige mecanismos de segurança a diferentes níveis, nomeadamente segurança do local remoto relativamente ao agente móvel, do agente relativamente ao local remoto, e segurança no próprio canal de comunicação. Por outro lado, e de uma perspectiva ortogonal à questão da mobilidade, pelo simples facto de se supor um espaço aberto de agentes, exige que os mesmos (ou pelo menos os seus donos) sejam devidamente certificados e/ou reconhecidos num determinado contexto.

(In) Existência de Normas: Para se poder criar espaços abertos de agentes, que sejam suportados por diferentes organizações, tipos de plataformas, etc., é fundamental que esses agentes possam vir a comunicar, a estabelecer redes de relações e de conhecimento de forma dinâmica. Para tal é necessário a definição de normas universalmente reconhecidas e utilizadas (tal como hoje o protocolo HTTP ou a especificação HTML para documentos o é). Tais normas impõem-se a diferentes níveis (alguns hoje mais fáceis de concretizar que outros):

- Ao nível dos protocolos de comunicação (e.g., HTTP, CORBA/IIOP, Java RMI);
- Ao nível das interfaces de programação (e.g., interfaces remotas em Java, IDL, ou MIDL);
- Ao nível da especificação sintáctica e semântica das mensagens (DTD em XML, tais como o *Commerce XML* (cXML) da *Ariba*, o CBL da *Commerce One*, *BizTalk* da *Microsoft*), mas, mais dificilmente, ao nível das ontologias de conhecimentos (XOL, KIF, OIL, ou RDF) que os agentes de alguma forma necessitam para poderem entender semanticamente os seus eventuais interlocutores.

2.10 Gestão Centralizada e Distribuída

No domínio das telecomunicações, a decisão de se implementar uma arquitectura de gestão de redes centralizada, hierarquizada, distribuída ou híbrida é um facto importante para a implementação efectiva, ou mesmo apropriada, de agentes de software.

O rápido crescimento das redes, associado à sua complexidade, flexibilidade e diversidade, tem levantado a grande questão sobre saber até que ponto um sistema centralizado é escalável. Infelizmente, a passagem de informação de monitorização, controlo e de confirmação desde os equipamentos de comutação até nas estações de gestão impõem um grande atraso no mecanismo de comunicação das camadas físicas básicas. Um sistema centralizado implica uma maior vulnerabilidade a colapsos quando um falha ocorre.

Não existe uma política localizada de controlo, efectuando-se o controlo através de um sistema de gestão central. No entanto, uma gestão centralizada fornece mecanismos efectivos. Por exemplo, um número pequeno de entidades redundantes requer uma activação controlada. Tal deve-se ao facto de uma aproximação centralizada oferecer

benefícios em termos do fornecimento de informação global do estado da rede, reduzindo a probabilidade de ocorrência de inconsistências e conflitos.

Obviamente que os benefícios oferecidos por este tipo de gestão devem ser pesados e ponderados face à necessidade de escalabilidade do sistema. Uma desvantagem óbvia da aproximação centralizada dos sistemas é a de não permitir que os dados de controlo sejam processados onde e quando tal é necessário. Por forma a diversificar aplicações e serviços de rede, devem ser considerados alguns pontos, quer haja uma necessidade de informação global propagada até aos variados pontos da rede, tais como a necessidade de disponibilizar interfaces comuns de gestão, quer a necessidade de reduzir gastos na distribuição geográfica da capacidade de processamento.

Contudo, o controlo das redes de telecomunicações é inerentemente distribuído e os sistemas de gestão necessitam de uma capacidade para tratarem excepções ocorridas em tempo útil de funcionamento das aplicações, e ser robustos sob essas circunstâncias. Na medida em que aumenta a complexidade dos serviços (particularmente nas redes de suporte a aplicações multimédia), maior é a quantidade de informação necessária de ser processada, e algumas falhas parciais no sistema requerem uma rápida rectificação. Por este motivo, a gestão de uma rede tem sido cada vez mais difícil de ser concretizada por mecanismos centralizados. Após a determinação destes factos, pode-se pensar no processo de gestão distribuído, através de, por exemplo:

- Distribuição de tarefas apropriadas e disponibilizadas pelos recursos computacionais;
- Melhoramento da robustez em termos de recuperação de erros e de tratamento de informação incerta;
- Melhoramento do tempo de resposta: o controlo localizado pode permitir respostas mais rápidas a eventos locais;
- Utilização de técnicas do domínio da inteligência artificial quando existe falta de conhecimento sobre como desenvolver um algoritmo de processamento óptimo, isto é, quando não existe um procedimento bem definido ou um algoritmo para tratar certos eventos;
- Possibilitar a extensão do fornecimento de novos serviços, como por exemplo, a introdução de novos serviços complexos como é o caso da difusão de vídeo na escolha para determinados grupos de utilizadores da rede (“*multicast video-on-demand*”).

2.11 Conclusões

Os agentes de software são melhor entendidos como um paradigma de interacção homem-máquina baseado no modelo de delegação de tarefas e/ou como uma tecnologia

de espectro largo, potencialmente aplicável em inúmeras áreas de aplicação. Dessa perspectiva, as principais conclusões a reter são que os agentes de software:

- Devem ser melhor caracterizados pelos seus atributos e benefícios correspondentes;
- Existem em diferentes áreas de aplicação: telecomunicações, comunicação social, gestão de empresas, comércio, etc. Não existem, regra geral, por si só, ou seja, encontram-se em sistemas e aplicações que os integram;
- Devem ser melhor entendidos como uma metáfora de concepção de aplicações que promove, entre outras, a delegação de tarefas, a personalização, a automatização, a monitorização de eventos, e a adaptabilidade a situações novas;

A designação de agente é utilizada num largo espectro de ambientes computacionais, de aplicações, e por variadas comunidades científicas. Este facto conduz necessariamente ao perigo de confusão que a sua designação implica.

Como conclusão, a adopção da tecnologia dos agentes na gestão e controlo de redes de telecomunicações está muito dependente da habilidade de se desenvolver uma “sociedade de agentes” que possa ser suficientemente flexível para responder aos requisitos futuros das referidas redes. O estado seguinte do desenvolvimento de agentes em sistemas de telecomunicações deverá precisar de mais estudos e está dependente das tecnologias de suporte na comunicação como é o caso, por exemplo, das linguagens de comunicação e das linguagens de conteúdo.

Assim, o impacto dos agentes no futuro será particularmente atraente e relevante caso se consigam ultrapassar alguns dos desafios e limitações identificados.

Capítulo 3

3 Arquitectura de Gestão

Actualmente o mercado das tecnologias de gestão de redes é caracterizado pela grande diversidade de aplicações, várias abordagens da gestão, pela tendência para a gestão descentralizada e pela implementação de soluções fechadas com o sistema de informação de gestão embebido.

Ao longo dos últimos anos, vários fabricantes de sistemas de gestão de redes, desenvolveram completos e complexos sistemas de informação para os vários ambientes de rede que pretendem gerir. Um dos problemas existentes destes sistemas de gestão reside no facto de os sistemas de informação usados serem proprietários e fechados.

A Arquitectura de Gestão de Redes MANIA pretende combater esses mesmos problemas, apresentando um modelo capaz de dar resposta a necessidades da gestão concretas como é o caso da gestão com recurso a múltiplos Sistemas de Gestão de Redes, a gestão de múltiplos domínios de gestão e de como gerar valor acrescentado.

Esta arquitectura alarga o conceito de gestão de redes além do SNMP e CMIP, através do uso de protocolos abertos, normalizados, e acima de tudo testados, como são o caso do HTTP, SMTP, IMAP, NFS, LDAP, entre outros, e de mecanismos de comunicação entre aplicações de gestão (Aplicações Autónomas, gestores intermédios, etc.).

3.1 Modelo Gestão de Redes

O desenvolvimento dos sistemas de gestão de redes tem sido efectuado em torno de quatro áreas fundamentais, conforme se pode ver na figura seguinte:

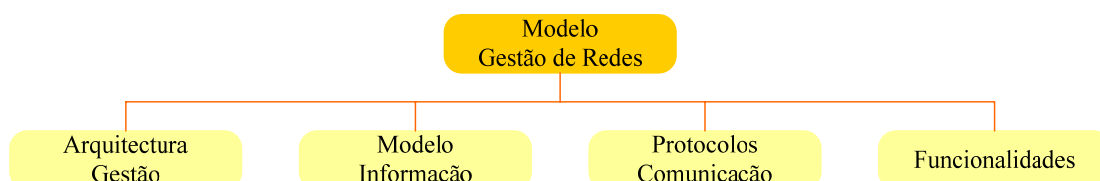


Figura 4 – Modelo de Gestão de Redes.

3.2 Arquitectura de Gestão

A arquitectura de gestão descreve, de uma forma geral, a estrutura das entidades envolvidas na gestão, as suas interfaces e métodos de comunicação.

Actualmente os elementos presentes nas arquitecturas de gestão de redes são quatro:

- Objecto Gerido;
- Sistema de Gestão de Rede;
- Protocolos de Gestão;
- Linguagens de Comunicação de Gestão.

3.2.1 Objecto Gerido

O Objecto Gerido é a abstracção de um recurso a ser gerido, recurso este que pode ser de comunicação, processamento de dados, serviço, utilizador ou outro, e que pode ainda ser físico ou lógico. Estes são representados através de uma MIB, a qual é trabalhada por um agente que lhe está associado, quer para a actualização da informação de gestão, quer na resposta a pedidos do sistema gestor.

3.2.2 Sistema de Gestão da Rede

Os sistemas de Gestão de rede são compostos também, à semelhança das Arquitecturas de Gestão, de Elementos, conforme se pode ver na figura seguinte.

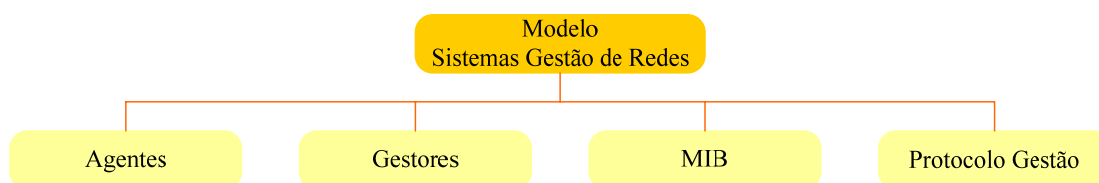


Figura 5 – Modelo Sistemas de Gestão de Redes.

3.2.2.1 Agentes

Os agentes são entidades que têm como objectivo recolher informação de gestão, armazenamento na MIB e comunicação com o sistema gestor.

Alguns agentes podem gerir objectos lógicos, tais como segmentos de rede, tráfego, etc. Nesse sentido, estes agentes são colocados em dispositivos apropriados, normalmente designados de sondas.

Como funcionalidades dos Agentes são de realçar:

- Implementar e manter os objectos MIB;
- Responder a operações do gestor;
- Gerar notificações, "traps" e informações;
- Implementar segurança;
- Definir políticas de acesso para gestores externos;
- Visualização da MIB (SNMPv3);
- Modo de acesso e gestão aos objectos MIB (SNMPv3).

Os agentes podem ser alojados em quase todos os dispositivos com capacidade de processamento, onde se incluem:

- Equipamentos Windows 2000/2003 ou superior;
- Equipamentos Linux/Unix;
- Estações e servidores Novell Netware;
- Equipamentos de rede, como hubs, switches, routers, etc.

3.2.2.2 Gestores

O gestor é a entidade que interage com os agentes.

No caso dos gestores temos como funcionalidades:

- Receber e enviar valores das instâncias dos objectos MIB nos agentes;
- Receber notificações dos agentes;
- Trocar mensagens com outros gestores.

O gestor é responsável pela comunicação e integração de todos os objectos geridos num sistema de informação e que os relaciona num contexto de domínio de rede.

3.2.2.3 MIB

A *Management Information Base* ou MIB, representa o repositório com a informação de gestão existente no Objecto Gerido, é um dos componentes fundamentais do sistema de gestão de redes, nomeadamente no que se refere ao conhecimento dos objectos geridos. Ela, contém a descrição dos dados de gestão, define a sintaxe e a semântica dos objectos geridos.

A MIB encontra-se organizada hierarquicamente e acessível através de um protocolo de gestão de redes, como por exemplo o SNMP. Assim, a informação de gestão é armazenada na Base de Informação de Gestão na forma de Objecto Gerido. O Objecto Gerido não é um objecto no sentido de "*object-oriented*", mas sim no sentido de representar um objecto variável.

As instâncias dos Objectos Geridos podem ser acedidas, através dos protocolos de gestão. É com base na MIB e utilizando o protocolo SNMP que gestores e agentes trocam assim instâncias de objectos geridos.

3.2.2.4 Protocolos de Gestão

Com base nos dois principais modelos conceptuais OSI e Internet, são vários os protocolos que se foram desenvolvendo. Apesar disto, dois têm sobrevivido ao longo dos tempos, são eles o SNMP em redes IP, com base no modelo Internet, e o CMIP nas redes de telecomunicações, baseado no modelo OSI.

SNMP – *Simple Network Management Protocol*

O protocolo SNMP é um protocolo que foi desenvolvido para a gestão de redes IP nos finais dos anos 80 e que se encontra já na sua terceira versão. O objectivo do SNMP é facilitar a troca de informação de gestão entre dispositivos de rede.

Tem como função efectuar, toda a comunicação de informação de gestão necessária, entre os agentes e gestores e, em alguns casos, entre gestores.

Permite aos gestores de rede gerir o desempenho, encontrar e resolver problemas na rede, bem como planear o crescimento da mesma. A utilização mais comum do SNMP encontra-se na gestão remota de dispositivos de rede. O SNMP permite aos construtores adicionar novas funções de gestão de rede aos seus produtos.

O SNMP foi desenhado sobre a pilha protocolar TCP/IP e não orientado à conexão numa filosofia de gestor/agente. Aqui, a estação de gestão recolhe a informação proveniente dos vários agentes, os quais podem ser implementados nos mais variados dispositivos.

A aplicação de gestão e os agentes efectuem as suas transacções com base nos comandos, *GetRequest*, *GetNextRequest*, *SetRequest*, *GetResponse* e *Trap*.

Estes comandos não foram sofrendo alterações na forma de utilizar, mas foram melhorados ao longo das versões, tendo sido igualmente adicionados novos comandos ao protocolo com o objectivo de responder às novas exigências como por exemplo o *GetBulk* e *Inform*.

A grande alteração entre a primeira e a segunda versão reside no aumento de funções do protocolo, tornando-se assim mais eficiente na troca de grandes quantidades de informação entre o gestor e o agente, e na comunicação entre gestores.

Na sua última versão, o SNMP vê assim corrigida a sua falha de segurança, alvo de grande crítica pelo mercado, e que, além de modificar parte da estrutura de comunicação, introduziu mecanismos de segurança.

CMIP – *Common Manegement Information Protocol*

Desenvolvido sobre o modelo OSI, o CMIP é um protocolo que define como a informação de gestão de redes é transmitida entre as aplicações de gestão e os agentes. O CMIP é um protocolo orientado à ligação, e possui funcionalidades de segurança, através de controlo de acesso, autorização e registo. A especificação do CMIP para redes IP é denominada CMOT – CMIP *over* TCP.

A troca de informação de gestão entre o gestor da rede e os agentes é realizada utilizando o conceito de Objectos Geridos. O Objecto Gerido é uma abstracção do dispositivo gerido o qual pode ser monitorizado, modificado, controlado e até ser utilizado para a execução de tarefas de gestão. A aplicação de gestão de redes pode iniciar assim uma transacção com o agente através dos seguintes comandos, *ACTION*, *CANCEL GET*, *CREATE*, *DELETE*, *GET*, *SET*, *EVENT REPORT*.

O CMIP não especifica funcionalidades do sistema de gestão de redes, só define o mecanismo de troca de informação de gestão dos objectos geridos e como a informação deve ser utilizada e interpretada.

CMIP vs SNMP

- As variáveis CMIP transportam informação e podem ser utilizadas para efectuar tarefas;
- O CMIP é um protocolo seguro;

- O CMIP permite efectuar mais acções que o SNMP num simples pedido;
- O CMIP possui melhores relatórios no que diz respeito a condições anormais da rede;
- O CMIP tem a desvantagem de requerer grande quantidade de recursos do sistema para gestão. Num ambiente onde os recursos disponíveis para a gestão são escassos faz com que a sua utilização seja condicionada;
- Por outro lado a sua complexidade torna difícil a programação, exigindo assim recursos humanos especializados e bem treinados para a sua implementação, manutenção e operação.

Estes requisitos, por parte do CMIP, tornam-no pouco prático para o ambiente de redes IP, ambiente este caracterizado por uma grande heterogeneidade de dispositivos e de expansão tecnológica e de serviços.

Na perspectiva aqui apresentada, temos pelo menos dois níveis de comunicação de gestão a serem considerados, são eles a comunicação entre os sistemas gestor e os objectos geridos e a um outro nível a comunicação entre sistemas gestores ou entre sistemas gestores e consolas de gestão.

Quanto ao primeiro nível, o SNMP tem vindo a assumir as suas funções e a responder às necessidades dos sistemas de gestão.

Ao nível da comunicação entre sistemas gestores e entre o sistema gestor e as consolas de gestão, habitualmente esta função é desempenhada em ambiente proprietário.

Na Arquitectura de Gestão de redes MANIA, a utilização de protocolos normalizados e a aceitação do ambiente de funcionamento Internet como uma realidade incontornável, faz com que o recurso aos protocolos já existentes ao nível da camada de aplicação seja uma mais valia para os sistemas de gestão de redes.

Protocolos como IMAP, SMTP, FTP, SMB, LDAP, entre outros, foram levados em consideração nesta nova arquitectura de gestão de redes.

3.2.3 Linguagens

Ao nível da comunicação entre agente e sistema gestor, os sistemas de gestão tradicionais já executam as suas tarefas de uma forma eficaz devido ao uso do SNMP, uma vez que este usa a linguagem ASN.1.

Quanto à comunicação entre sistemas gestores, e entre sistemas gestores e consolas, foi necessário repensá-la e defini-la, pois a realidade actual não contempla estas necessidades emergentes, visto o desenvolvimento se basear na comunicação interna e de uma forma proprietária.

Para a comunicação entre sistemas gestores e entre sistemas gestores e consolas, a linguagem assenta no HTML, com especial interesse no XML, pela sua versatilidade na representação e transporte de informação [21].

A linguagem XML, como subconjunto da linguagem SGML, é usada para a representação de dados. Por exemplo a informação de gestão em formato texto. Uma

das vantagens do XML prende-se com a possibilidade de, opcionalmente, incorporar-se a própria gramática em anexo, permitindo assim enviar dados e a estrutura dos dados utilizada. A gramática para um documento XML é descrita através do mecanismo conhecido como DTD que é responsável por descrever os elementos num documento XML. Para que a informação de gestão possa utilizar convenientemente o XML é necessário definir um vocabulário e DTDs.

Os documentos XML contêm informação, cuja forma de captar ou de a mostrar pode ser alcançada com o uso de documentos XSL. Assim, um documento XML associado a um XSL pode ser utilizado para a apresentação e representação gráfica da informação contida no documento.

A definição de DTDs, conjuntamente com as capacidades do XSL, fornece uma poderosa forma de comunicar informação de gestão em ambientes heterogéneos, utilizando o HTTP.

3.3 Modelo de Informação

A informação de gestão do elemento encontra-se organizada e a MIB é a forma normalizada mais utilizada em ambiente Internet. O aparecimento ou desenvolvimento de novos elementos não é problemático pois a possibilidade de gerar e disponibilizar a sua MIB é um processo já utilizado e a funcionar.

Enquanto que a informação de gestão de elementos encontra-se normalizada, quer ao nível da semântica, sintaxe, linguagem e protocolo de comunicação, a informação de gestão de rede ainda não obteve consenso por parte dos construtores.

A informação de gestão da rede é complexa, não está normalizada e encontra-se no interior do sistema gestor num formato normalmente proprietário, quer em termos de estrutura, quer de conteúdo. A identificação da topologia da rede é um exemplo deste tipo de informação.

Para que a comunicação entre sistemas gestores seja possível é necessário que estes conheçam e reconheçam a informação utilizada e disponibilizada por cada um deles. A única forma de o fazer é recorrendo à normalização. Normalizar o significado (semântica), normalizar a forma (sintaxe) e normalizar a comunicação (linguagem) são os três principais passos para que a informação de gestão de redes possa ser utilizada em ambiente aberto e de uma forma transparente.

Reconhecendo que é uma área de difícil consenso e complicada, é também reconhecida como uma área crucial para que a comunicação entre sistemas gestores heterogéneos possa ser realizável.

A possibilidade de integrar transparentemente dois ou mais destes sistemas, de forma a se complementarem, interagirem, ou simplesmente disponibilizarem informação tratada, é assim remota. Daí a grande importância e interesse na criação de uma estrutura de comunicação que permita a troca transparente de informação de gestão de rede entre os mais variados sistemas gestores.

3.3.1 Sistema de Informação de Gestão de Redes

A par do ambiente de comunicação, com os seus protocolos e linguagens, a localização e a forma como a informação de gestão é armazenada e acessível é de importância cada vez mais crítica. Por este motivo, não só é importante separar a consola do sistema de gestão e torná-la independente, como também é fundamental que a informação de gestão esteja acessível na rede e não, como é actualmente, propriedade do sistema de gestão.

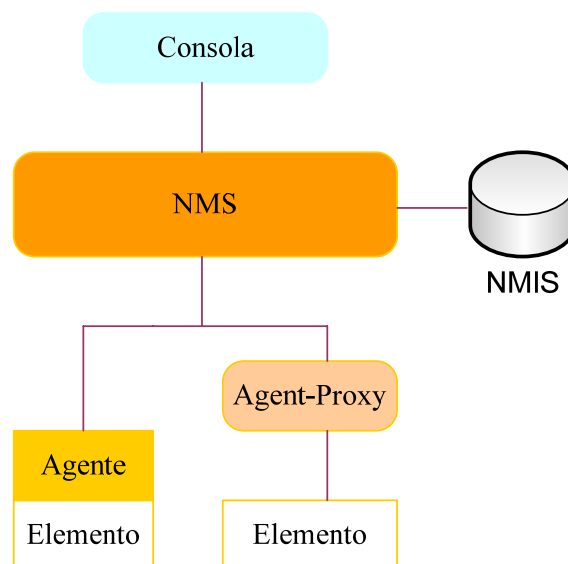


Figura 6 – Sistema de Informação de Redes.

O sistema de informação de gestão de redes deve conter um vasto conjunto de informação que, no que diz respeito à gestão de elementos na rede, pode ser agrupada da seguinte forma:

- **Informação dos Elementos de Rede** – informação que caracteriza cada um dos elementos de rede, susceptíveis de serem geridos, ou pelo menos monitorizáveis. Estes elementos podem ser físicos ou lógicos, são passíveis de serem utilizados nesta, ou em qualquer outra rede, pelo que esta informação deverá ser universal;
- **Informação dos Objectos da Rede** (Instanciação dos Elementos de Rede) – informação sobre cada objecto gerido da rede, necessário para identificar o objecto em causa, a sua localização, quer geográfica como topológica, e saber o seu estado;
- **Informação da Topologia** – informação sobre a forma como cada elemento se interliga na rede, suas dependências, estados e forma de acesso;
- **Informação de Eventos** – informação sobre as ocorrências na rede, esta informação pode ser obtida directamente a partir dos objectos geridos, ou através de serviços de correlações de eventos;
- **Informação do Domínio de Gestão** – informação sobre o domínio de gestão, sistemas gestores, interacções entre eles e relações com outros domínios.

Parte desta informação tem interesse exclusivo para a gestão no domínio (intra-domínio), outra parte para a gestão inter-domínio.

Quando inserido num domínio de rede, o administrador tem necessidade de conhecer como a rede está estruturada, quais são os elementos que a constituem, o seu estado e como podem ser utilizados.

Esta informação é relevante para a gestão e manutenção da rede, mas também para a disponibilização de serviços, o controlo de segurança, a gestão de utilizadores, etc.

De uma forma equivalente, a informação de utilizadores, de serviços, de políticas, etc., é relevante para a gestão de elementos.

Assim, de forma a ir ao encontro dos objectivos de gestão de redes é necessário obter e tratar este conjunto de informação, que, pela sua especificidade e importância, deve ser devidamente organizada e acessível aos sistemas gestores do domínio em questão. Esta informação pode ser tratada de acordo com a sua volatilidade, dinâmica ou estática. Neste contexto, surge a importância de definir a informação de gestão da rede e de como esta deve estar organizada e disponível na rede, não só no contexto de gestão dos elementos de rede mas também no dos serviços, políticas, sistemas, utilizadores, negócio, parque, etc.

3.3.2 Barramento de Informação de Gestão de Redes

O conceito de barramento de informação de gestão de redes aqui apresentado, tem por objectivo funcionar como plataforma de comunicação comum, de forma a possibilitar a comunicação transparente e normalizada entre dois ou mais sistemas gestores pertencentes ao mesmo domínio de gestão.

A informação a trocar através deste barramento é informação cuja relevância para a gestão é elevada e com uma validade temporal relativamente grande, como seja toda a informação de identificação dos objectos geridos existentes no domínio de gestão da rede, a topologia da rede, os utilizadores, os sistemas e toda a estrutura de meta-informação associada à gestão em tempo real, como seja a forma de aceder aos eventos, alarmes, estado, etc.

A informação de gestão da rede, dos serviços, dos utilizadores, do parque e de políticas são exemplos de informação que embora trabalhada por aplicações de gestão distintas, poderão partilhar a informação através deste barramento e assim, evitar a repetição de tarefas (Figura 7).



Figura 7 – Bus de Informação de Gestão de Redes baseado no LDAP.

Para a implementação do conceito foram avaliadas as tecnologias existentes, e o recurso aos sistemas de directórios, mais concretamente o LDAP, para suportar este conceito foi o elegido, pela sua estrutura, versatilidade e aceitação no mercado.

Posto isto, o recurso aos sistemas de directórios e a utilização do LDAP para o armazenamento e acesso à informação de gestão da rede, nomeadamente no que diz respeito à identificação dos recursos existentes na rede, a forma de lhes aceder, quer para monitorização, como para manutenção e configuração, foi a opção tomada.

A simplicidade de representação da informação de gestão em linhas de texto e a possibilidade de representação de árvores de informação é a base do sistema de informação de rede neste contexto.

Desta forma é possível identificar uma árvore de directório para suportar a estrutura do barramento de informação de gestão da rede (Figura 8). A estrutura base para a árvore de informação de gestão de rede tem duas vertentes de utilização:

- Utilização intra-domínio, cuja informação é integralmente partilhada e disponível aos sistemas gestores;
- Utilização inter-domínio, cuja informação partilhada não diz respeito ao conteúdo do domínio num primeiro nível, mas sim aos sistemas gestores disponíveis e como estes podem ser contactados para funções de gestão inter-domínio.



Figura 8 – Árvore de Directório de Informação de Gestão de Rede.

Da árvore apresentada, a informação do domínio e dos sistemas gestores é de carácter público, enquanto que a restante informação, e a que se encontra dependente dos sistemas gestores é de carácter privado do domínio.

A informação contida assim no sistema de directórios que contém a informação de gestão de rede do domínio é fundamentalmente estática ou meta-informação de gestão de redes. Esta informação diz respeito a toda a informação de carácter mais permanente que identifica e caracteriza todo ambiente do domínio da rede. São exemplo deste tipo de informação a caracterização dos elementos e a topologia de rede. Como exemplo de meta-informação tem-se a definição da localização da informação sobre os eventos, do

estado dos objectos geridos (as instâncias dos elementos de rede existentes no domínio), a localização das MIBs, etc.

Normalmente esta informação é de longa duração, a qual é consultada com bastante frequência, mas que está sujeita a poucas alterações, e é assim estruturada e armazenada em sistema de directórios, daí a necessidade de recorrer aos Sistemas de Directório, mais concretamente o LDAP.

Tendo acesso a esta informação de gestão de rede, existe ainda uma vasta franja de informação que fica fora desta estrutura, como por exemplo, os eventos, alarmes e outros, que ocorrem com uma regularidade bastante elevada e que apesar da sua grande importância para a gestão, têm um valor de oportunidade bastante limitado no tempo.

Esta informação, dinâmica, a qual tem uma importância mais relativa e que se desvanece no tempo, é proposta permanecer na base de dados interna da aplicação. A forma de aceder do exterior a esta informação, é por exemplo através da colocação da mesma acessível via IMAP. Assim, a meta-informação para o acesso efectivo a esta informação deverá encontrar-se no Sistema de Directórios, e cujo caminho deve ser introduzido em formato XML e acessível por exemplo via IMAP.

3.3.3 LDAP – Lightweight Directory Access Protocol

O *Lightweight Directory Access Protocol* é um protocolo que funciona sobre a pilha protocolar TCP/IP, no modelo cliente-servidor, utilizado para aceder a um serviço de Directório. Um Directório é uma base de dados especializada e optimizada para leitura, pesquisa e expansão.

Os Directórios contêm informações descritivas, baseadas em atributos, e são organizados em forma de árvore e não na forma de tabelas. A informação num Directório é geralmente mais lida do que é escrita. Como consequência, os Directórios, normalmente não são usados para implementar transacções complexas, ou consultas regulares em bases de dados, transacções estas que são usadas para fazer um grande volume de actualizações complexas. As actualizações nos Directórios são tipicamente simples ou nem são feitas.

Os Directórios estão preparados para dar uma resposta rápida a um grande volume de consultas ou operações de pesquisa. Eles podem também, ter a habilidade de replicar a informação, isto é usado para acrescentar disponibilidade e fiabilidade, enquanto reduzem o tempo de resposta.

Existem várias maneiras para disponibilizar um serviço de Directório. Métodos diferentes permitem que diferentes tipos de informação possam ser armazenadas no Directório, colocando requisitos diferentes, sobre como a informação pode ser referenciada, requisitada, actualizada, protegida de acessos não autorizados, etc. Alguns serviços de Directório são locais, fornecendo o serviço para um contexto restrito (ex., o serviço *Finger* numa máquina isolada). Outros serviços são globais, fornecendo o serviço para um contexto muito maior (por exemplo, a própria Internet).

Serviços globais normalmente são distribuídos, ou seja, cada servidor é responsável por uma parte apenas. O serviço de resolução de nomes, o DNS, é um exemplo, ele é um tipo de serviço de Directório, embora bastante especializado.

3.3.3.1 Modelo de Informação no LDAP

O modelo de informação do Serviço do Directório LDAP é baseado em entradas. Uma entrada é um conjunto de atributos, e é referenciada através de um nome distinto – DN [22]. O DN é usado para referenciar uma entrada de forma não ambígua. Cada um dos atributos das entradas tem um tipo e um ou mais valores.

Estes tipos geralmente são palavras mnemónicas, como CN para nome comum, ou MAIL para endereço de correio electrónico. Existem RFCs – *Request For Comments* que determinam estas palavras.

No LDAP, as entradas são organizadas numa hierarquia em árvore, tradicionalmente, essa estrutura reflecte limites geográficos e/ou políticos. As entradas que representam países aparecem no topo da árvore, abaixo dessas estão as entradas que representam estados/regiões e organizações nacionais. Abaixo podem estar entradas a representar unidades organizacionais, pessoas, impressoras, documentos ou qualquer outra informação que se deseje. A árvore também pode ser organizada com base em nomes de domínio.

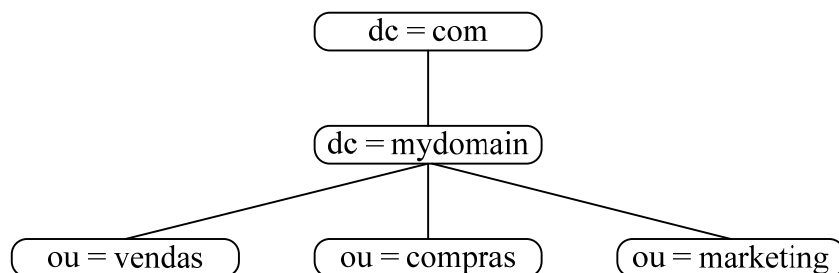


Figura 9 – Exemplo de árvore de informação LDAP.

Apesar de termos entradas para países, o Directório não possui uma entidade reguladora ou centralizadora como, por exemplo, os *root servers* do DNS. A separação por países, por exemplo, pode ser útil para empresas multinacionais.

É possível criar qualquer tipo de atributo, embora no LDAP existem diversas classes de objectos, e cada classe contém uma lista de atributos obrigatórios e opcionais. Essa lista é definida num RFC [22].

O servidor LDAP pode ser configurado para verificar as classes e forçar o uso correcto dos atributos. Quando se define uma entrada no Directório da classe `person`, um atributo `description` será opcional. As entradas nos Directórios podem ter várias classes diferentes, basta apenas observar os requisitos de atributos de cada classe.

Uma entrada é referenciada pelo seu DN, que é construído pela concatenação do nome da entrada, Nome Distinto Relativo (RDN) e de todas as entradas anteriores. É possível fazer uma comparação com *hostname* (RDN) e FQDN (DN).

3.3.3.2 Funcionamento no LDAP

O serviço de Directório LDAP é baseado no modelo cliente-servidor. Um ou mais servidores LDAP contêm os dados, criando a árvore do Directório LDAP. Um cliente LDAP liga-se a um servidor e faz um pedido. O servidor responde ao pedido, ou então responde com um ponteiro para onde o cliente pode conseguir a informação (tipicamente, outro servidor LDAP). É possível fazer novamente uma comparação com o DNS, a diferença é que o servidor LDAP não faz buscas recursivas, ou seja, em nome do cliente. O cliente é encarregado de procurar pelo servidor até encontrar a informação desejada.

O LDAP define operações para consultar e actualizar o Directório. As operações permitem a adição e remoção de uma entrada do Directório, modificação de uma entrada existente e modificação do nome de uma entrada. A operação de pesquisa pode abranger toda a árvore ou apenas um ramo, sem descer ou subir para os restantes.

Além de se especificar com filtros quais as entradas a encontrar, também é possível especificar quais os atributos a serem pesquisados. Se os atributos não forem especificados, vão ser todos retornados na resposta.

Alguns serviços de Directório não fornecem protecção contra acessos não autorizados, permitindo desta forma que qualquer um possa ver as informações. O LDAP fornece um método para autenticação de um cliente.

3.3.4 Directório de Informação de Gestão de Redes do domínio

Um dos aspectos mais relevantes para a gestão de redes em domínios com múltiplos sistemas gestores ou para a gestão inter-domínio prende-se com a localização da informação e de como lhe aceder.

Assim, é importante ter à disposição na rede a informação de gestão mais relevante, para que em qualquer altura seja possível efectuar as tarefas de gestão necessárias.

A transformação do modelo do sistema de gestão, por si só não implementa esta funcionalidade. O modelo possibilita o recurso ao sistema gestor e à informação de gestão da sua responsabilidade de uma maneira mais normalizada e aberta, mas não indica qual é a informação que dispõe nem como o domínio é gerido.

Assim surgiu o directório de informação de gestão do domínio, NMID, baseado no LDAP [23] [24] [25]. Este elemento deve estar presente em todos os domínios geridos, um por domínio e pode ser referenciado a partir do DNS.

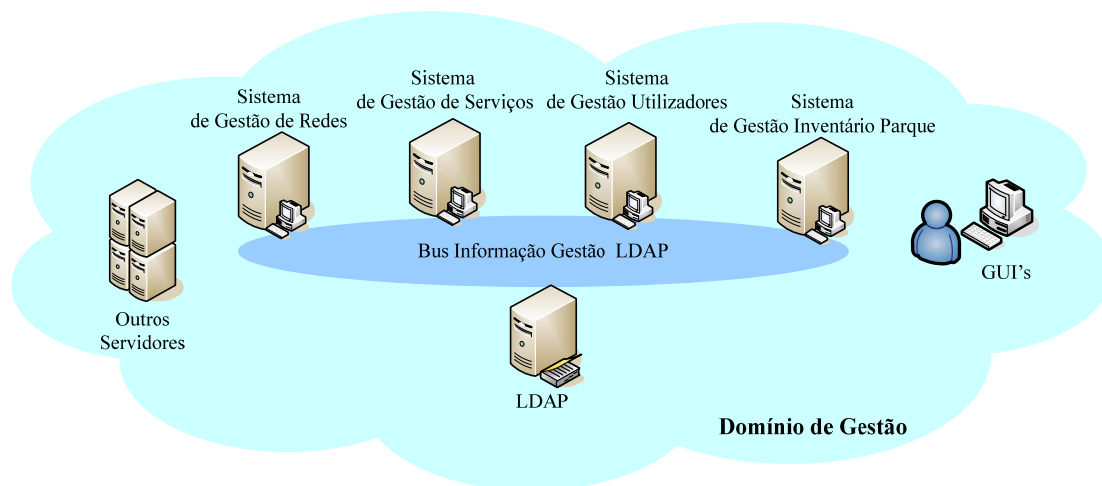


Figura 10 – Domínio de Gestão da Rede.

Desta forma os sistemas gestores não necessitarão guardar toda a informação no seu interior, partilhando assim informação de gestão, que por outros meios seria muito complicado (Figura 10).

Por outro lado, a rede passa a ter um local onde é possível ter informação sobre os sistemas gestores existentes no domínio, o que eles fazem, onde e como é possível aceder-lhes, bem como informação sobre os elementos mais relevantes e partilháveis de cada um.

Este elemento além de facilitar a melhorar a gestão dentro do domínio de gestão, tem também como função a gestão inter-domínio, possibilitando assim a gestão partilhada de recursos e sua monitorização.

3.4 Protocolos Comunicação Gestão

Na perspectiva tradicional, existem pelo menos dois níveis de comunicação de gestão a serem considerados:

- Comunicação entre os sistemas gestor e objectos geridos;
- Comunicação entre sistemas gestores ou entre sistemas gestores e consolas de gestão.

Quanto ao primeiro nível, o SNMP tem vindo a assumir as suas funções e a responder às necessidades dos sistemas de gestão.

Ao nível da comunicação entre sistemas gestores e entre o sistema gestor e as consolas de gestão, habitualmente esta função é desempenhada em ambiente proprietário.

Neste modelo, a utilização de protocolos normalizados e a aceitação do ambiente de funcionamento Internet como uma realidade incontornável, como o IMAP, SMTP, FTP, SMB, LDAP, faz com que o recurso aos protocolos já existentes ao nível da camada de aplicação seja uma mais valia para os sistemas de gestão de redes.

Nesta nova configuração do modelo de arquitectura de gestão, a comunicação, que antes era só analisada na perspectiva gestor – agente, assume aqui um novo papel, que é

atribuído pela utilização da consola e do repositório de informação de gestão de redes de uma forma autónoma e modularizada.

Mantendo toda a estrutura de comunicação e gestão entre gestores e agentes definidas nos modelos tradicionais, foi necessário redefinir a comunicação entre gestores e entre estes e as consolas (Figura 11).

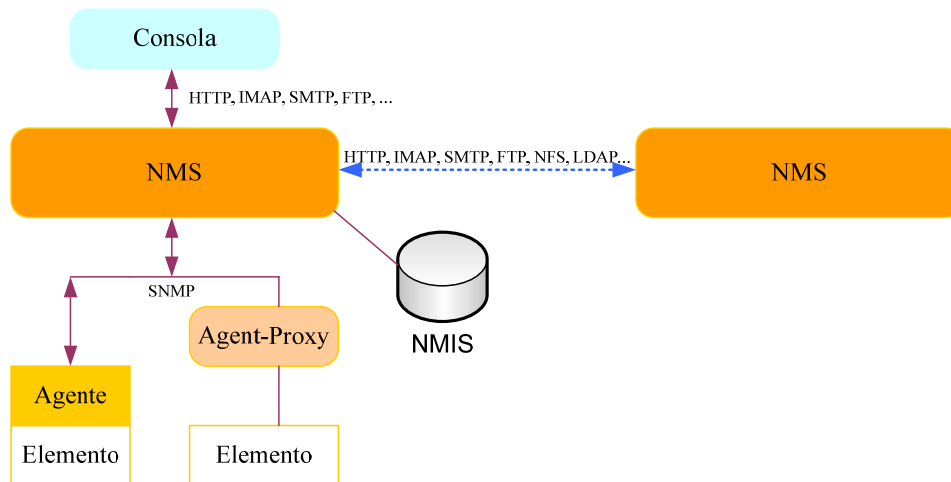


Figura 11 – Protocolos de Comunicação no Ambiente de Gestão.

Dado não existir ainda nenhuma normalização quanto à comunicação de gestão, a este nível, optou-se pela utilização dos protocolos normalizados e mais utilizados no âmbito da Internet. Neste modelo, a comunicação em ambiente de gestão é equivalente à das outras actividades, utiliza o mesmo meio, pelo que os protocolos existentes podem perfeitamente servir de base à actividade de comunicação de gestão entre sistemas gestores e entre estes e as consolas de gestão.

Desta forma, a comunicação em ambiente de gestão passa a ser suportada a dois níveis:

- A comunicação entre agente e sistema gestor, fazendo uso do SNMP ou CMIP e todos os seus mecanismos, já utilizados e testados;
- A comunicação entre sistemas gestores e entre sistemas gestores – consolas, através de protocolos de comunicação normalizados para a Internet, nomeadamente o http, IMAP e SMTP. Existem já alguns desenvolvimentos feitos a este nível, mas o mercado ainda encontra-se reticente, não tendo sido até à data assumido qualquer tipo de normalização.

A Arquitectura de Gestão de Redes MANIA, pretende alterar a arquitectura dos sistemas de gestão actuais, de forma a manter o que é bom nos sistemas actuais, comunicação do agente – sistema gestor e os serviços de gestão, e normalizar e autonomizar o sistema de informação de gestão de redes que vai além do Objecto Gerido, tratado pelo SNMP e com o recurso à MIB, bem como criar um nível de comunicação de gestão entre sistemas gestores e entre estes e as consolas (Figura 12).

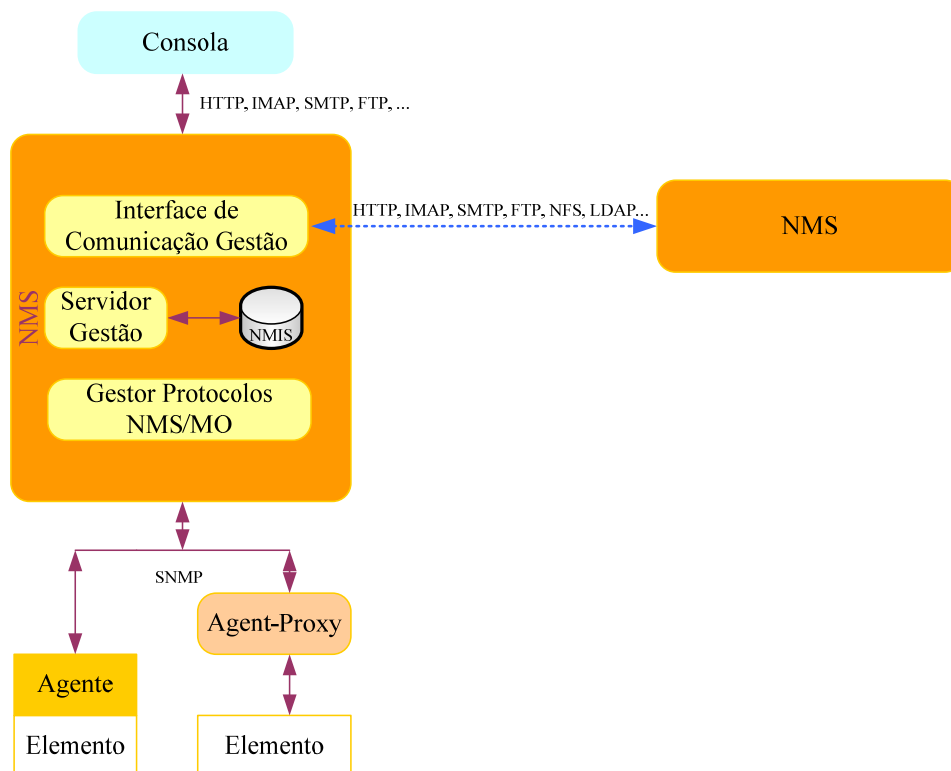


Figura 12 – Arquitectura Sistema Gestão.

Nesta arquitectura, o NMS é constituído por um módulo de gestão de comunicação entre o sistema gestor e os agentes (Gestor de Protocolos NMS/MO). O coração é o servidor de gestão, o qual contém todo o *know-how* da gestão de redes e coloca-o em prática na gestão da rede. A informação de gestão deverá ser armazenada num sistema de informação flexível e versátil, o NMIS. Por fim, o NMS necessita uma interface de comunicação de gestão que possibilite comunicar com as consolas e outros NMSs de uma forma o mais aberta possível e normalizada para o ambiente em que se desenvolve a gestão.

3.5 Funcionalidades

As funções do sistema de gestão correspondem ao conjunto de tarefas necessárias para a manipulação dos Objectos Geridos com o objectivo de executar acções sobre eles mesmos de acordo com estratégias e planos de gestão dos domínios de rede em que se encontram integrados. São exemplo as funções: criar, apagar, examinar, modificar objectos geridos.

Igualmente como exemplo encontram-se as funções de gestão do estado dos objectos geridos, as quais colocam à disposição do gestor a capacidade de examinar as alterações de estado, monitorizar sobrecargas, etc. Recorrendo ao modelo que representa o ambiente de gestão a quatro dimensões [1] (Figura 13) encontram-se aqui representadas as funções necessárias para desempenhar as operações comuns sobre os objectos geridos.

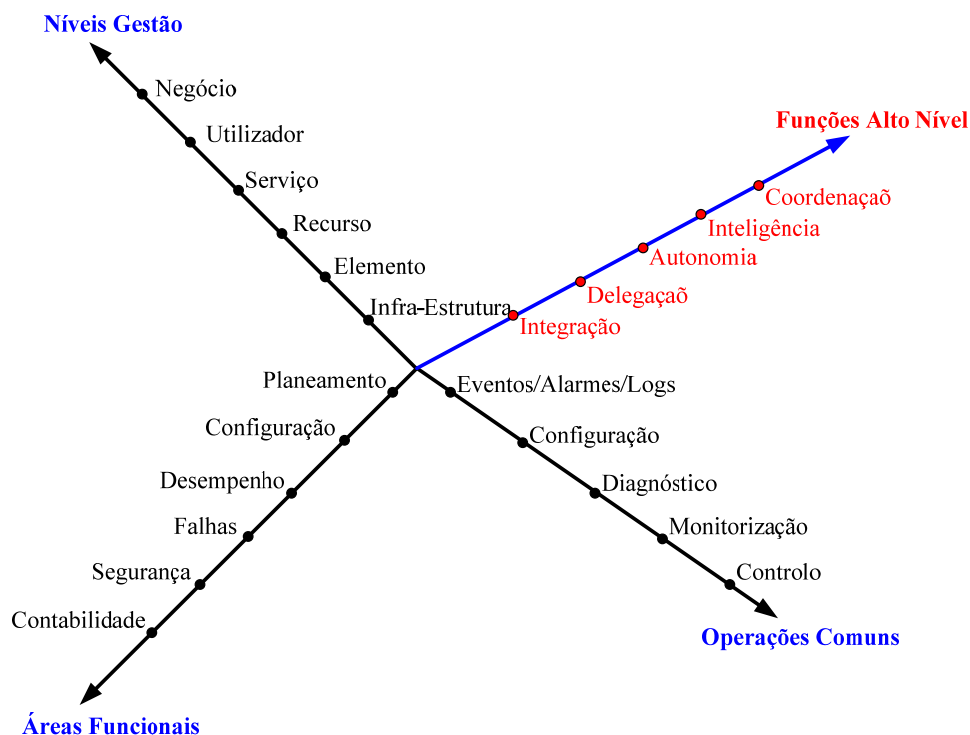


Figura 13 – Dimensões da Gestão.

Esta figura apresenta as quatro dimensões da gestão de redes. Uma mais operacional onde as tarefas básicas de gestão se encontram enumeradas, outra mais funcional descrevendo as funções de gestão, outra, referente aos domínios de intervenção da gestão e por último a dimensão que reflecte o nível de relação do sistema de gestão com todos os seus parceiros, directos e indirectos, pares ou complementares, humanos ou computacionais.

O que distingue os MNS, uns dos outros é a capacidade que estes têm em reflectir a realidade, bem como os mecanismos que permitem intervir sobre ela de uma forma o mais pró activa possível. A diferença entre a imagem obtida através do NMS e a situação real deve ser a mínima possível. O grau de sucesso atribuído a um NMS é assim directamente proporcional à diferença.

As exigências crescentes relacionadas com a alteração permanente da estrutura das redes, quer em dimensão como em heterogeneidade de dispositivos e associada à evolução tecnológica tem obrigado a repensar as plataformas de gestão, e desta forma a exigir que estas possuam funções de alto nível incorporadas, nomeadamente a de integração e delegação.

Uma curiosidade neste domínio dos NMSs generalistas é o facto de que o *OpenView* da HP, o *Tivoli NetView* da IBM, e o *Unicenter* da Computer Associates monopolizam o mercado. E mais interessante é que a integração é uma característica apresentada pelos três fabricantes, mas com uma pequena nuance, esta funcionalidade aplica-se entre produtos do mesmo fabricante e mesmo segmento, ou entre produtos complementares e através de APIs bem determinados.

3.6 Vantagens

Esta arquitectura tem a vantagem de poder reutilizar grande parte, se não a totalidade, do *know-how* e de todo o desenvolvimento efectuado até agora, permitindo aos NMSs alargar o horizonte operacional às funções de alto nível.

Nesta arquitectura do sistema de gestão de redes encontram-se os habituais módulos na camada de servidor. A comunicação com os agentes continua a processar-se no ambiente tradicional, com base no SNMP.

A camada de gestor, por seu lado, é dividida em duas:

- Camada de gestor;
- Camada de interface gráfico de gestão.

Esta última resulta da definitiva separação das funções de consola do sistema gestor e deverá funcionar de uma forma aberta e com base em protocolos normalizados para a Internet.

A camada de gestor passa a contemplar uma interface de comunicação de gestão de redes, responsável por efectuar a comunicação entre o sistema gestor e a consola, bem como com outros sistemas gestores, tendo por base protocolos de comunicação normalizados e a funcionar no ambiente da Internet conforme referido anteriormente.

A colocação da informação de gestão de rede neste módulo vai permitir que, através de simples procedimentos de comunicação entre sistemas gestores ou de pedidos via consola, seja mais rápido e eficaz o acesso à informação. A separação entre informação estática e informação dinâmica é de extrema importância para o desenho e utilização dos mecanismos de armazenamento e consulta da mesma.

Assim, o sistema gestor deverá dispor de um conjunto de interfaces para a utilização dos protocolos de comunicação mais utilizados, tais como HTTP, IMAP, SMTP, FTP, LDAP, SMB, etc. Sobre estes, com o recurso à linguagem XML/XSL, é possível comunicar de uma forma semelhante, tanto com a consola, como com outros sistemas gestores e desta forma partilhar ou monitorizar a informação de gestão dos recursos geridos.

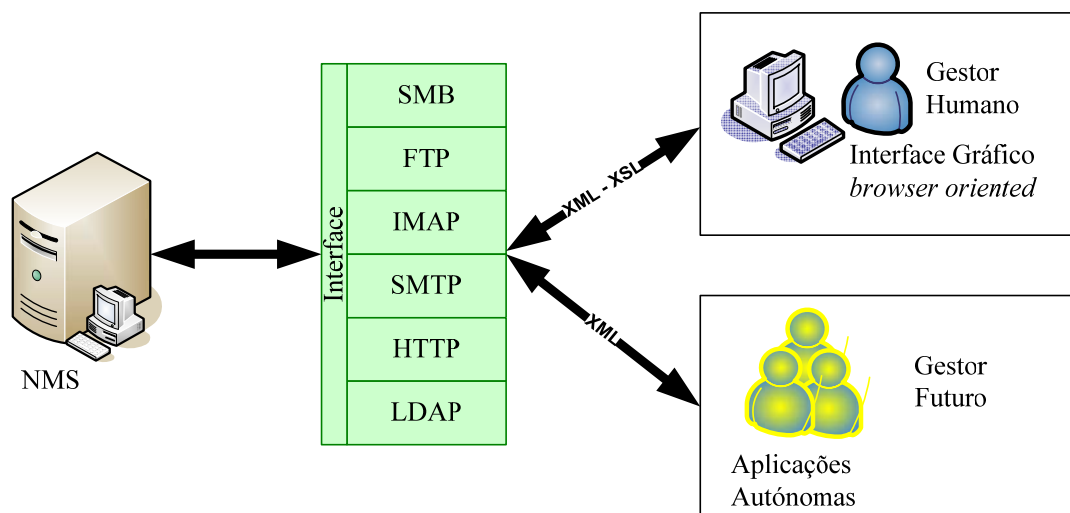


Figura 14 – Interfaces para protocolos de comunicação.

3.7 Conclusões

Este modelo, é um dos caminhos possíveis, para permitir aos sistemas de gestão de redes subir nas funções de gestão de redes de alto nível, devido à capacidade de comunicação entre sistemas gestores, a autonomização da consola de gestão face ao sistema gestor e a possibilidade de aceder à informação de gestão na rede, com base em protocolos normalizados.

Esta nova arquitectura pretende possibilitar uma gestão mais distribuída, disponibilizar a informação de gestão na rede, directamente se for informação estática, ou através de meta-informação se for mais dinâmica ou dependente do sistema gestor que a manipula.

Por outro lado, o recurso às normas já existentes na Internet para a comunicação de aplicações, à utilização do XML/XSL como linguagem, e à utilização do sistema de directórios como suporte à informação de gestão de rede do domínio de gestão são os factores que, em conjunto, permitirão a este modelo incorporar valor acrescentado aos actuais e tradicionais sistemas de gestão.

A utilização do conceito de barramento de informação de gestão de redes e do directório de informação de gestão é a base para permitir aos sistemas de gestão, sejam eles quais forem, comunicarem, interagirem, partilharem e delegarem funções.

Uma vez que a informação de gestão (rede, serviço, utilizadores, negócio, inventário e parque informático), está disponível e acessível através da rede, de uma forma aberta e normalizada, é possível afirmar que está aberto o caminho para os futuros decisores de gestão, as Aplicações Autónomas.

Capítulo 4

4 Arquitectura de Comunicação do Sistema de Aplicações Autónomas de Gestão

Apresentada a Arquitectura de Gestão, é necessário falar agora sobre as Aplicações Autónomas. Vamos ver como será possível, através de tecnologias normalizadas, diferentes aplicações de gestão comunicarem e partilharem a sua informação de forma a automatizarem os aspectos manuais e rotineiros da gestão de uma infra-estrutura de Tecnologias de Informação.

Para alcançar este objectivo, é proposta uma Arquitectura Multi-Agente [26], que pretende integrar diferentes aplicações de gestão existentes, utilizando *Instant Messaging*, mais concretamente o protocolo XMPP.

Por outro lado, esta arquitectura é baseada no inovador modelo global de informação baseado no LDAP ou LMIB [27].

Este modelo global de informação, foi desenvolvido para diferentes aplicações de gestão partilharem a sua informação de uma forma simples e normalizada, sem a necessidade de recorrer a uso de interfaces ou APIs proprietárias.

4.1 Introdução

É numa perspectiva de necessidade de integração e partilha da informação de gestão, que esta nova arquitectura pretende dar resposta.

A comunicação num ambiente aberto e heterogéneo entre Sistemas de Gestão, que podem ser responsáveis por várias áreas como, por exemplo gestão de Elementos, Serviços, Utilizadores, Negócio, Apoio, e até Aplicações Autónomas) é uma necessidade com vista para a delegação e a coordenação da gestão, ou seja, a incorporar as funções de gestão de alto nível (Figura 15).

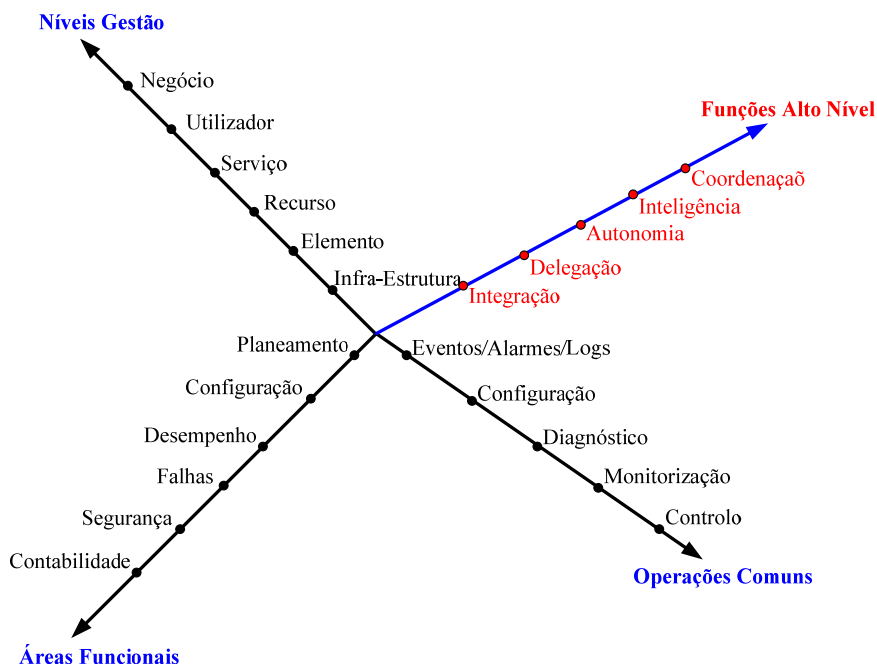


Figura 15 – Dimensões da Gestão.

Sendo assim, é necessária uma plataforma de informação. Esta, vai ser responsável por permitir aos Utilizadores, Aplicações Autónomas e Aplicações de Gestão partilharem a sua informação de uma forma aberta e normalizada.

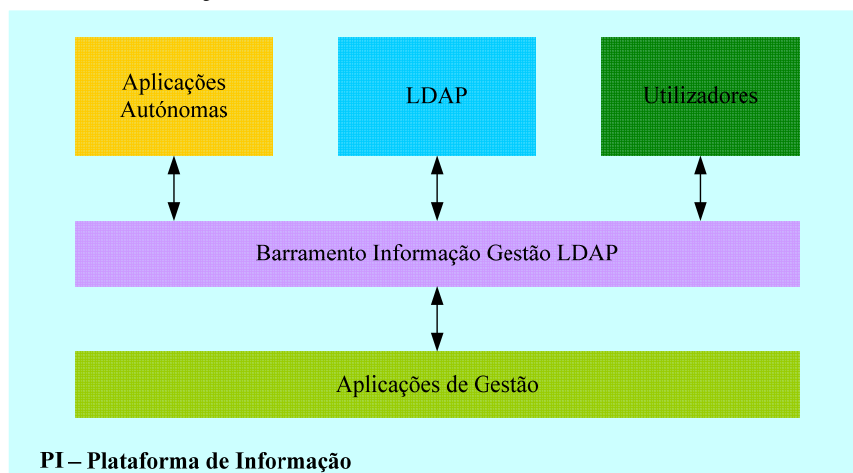


Figura 16 – Plataforma de Informação.

Conforme é visível na figura anterior, a plataforma de informação é composta por Aplicações Autónomas, servidor LDAP, Utilizadores, um Barramento de Informação de Gestão LDAP e por fim as Aplicações de Gestão.

Os Utilizadores são os administradores ou operadores de redes, as Aplicações Autónomas podem ser encaradas como entidades que possuem uma série de capacidades e objectivos definidos. Elas interagem com o ambiente em que estão inseridos e com outros agentes. Através destas interacções são capazes de realizar tarefas de gestão.

O barramento de informação de gestão LDAP tem por objectivo funcionar como plataforma de comunicação comum, de forma a possibilitar a comunicação transparente e normalizada entre Aplicações Autónomas, Aplicações de Gestão e Utilizadores, pertencentes a um domínio de gestão.

A informação a trocar através do barramento de informação de gestão LDAP é informação muito relevante para a gestão como a informação de topologia da rede, utilizadores, sistemas, serviços e toda a estrutura de meta-informação associada à gestão em tempo real, como seja a forma de aceder aos eventos, alarmes, etc.

A informação do Sistema de Gestão de Redes, do Sistema de Gestão de Serviços, do Sistema de Gestão de Utilizadores, do Sistema de Gestão de Inventário do Parque são exemplos de informação, que embora trabalhada por aplicações de gestão distintas, poderão partilhar a sua informação através deste barramento e assim, evitar a repetição de tarefas.



Figura 17 – Modelo Informação Global LDAP.

Para a implementação do barramento de informação de gestão LDAP, a escolha recaiu sobre os sistemas de directórios, que conforme o próprio nome indica, o LDAP foi o elegido pela sua estrutura, versatilidade e aceitação no mercado.

Assim, o recurso aos sistemas de directórios e a utilização do LDAP para o armazenamento e acesso à informação de gestão, nomeadamente no que diz respeito à identificação dos recursos existentes na rede, a forma de lhes aceder, quer para monitorização, como para manutenção e configuração, foi a opção tomada.

A simplicidade de representação da informação de gestão em linhas de texto e a possibilidade de representação de árvores de informação é a base do sistema de informação de rede neste contexto. É desta forma possível identificar uma árvore de directórios para suportar a estrutura do barramento de informação de gestão da rede.

4.2 Modelo de Informação

Para que a comunicação seja possível é necessário que as Aplicações de Gestão e as Aplicações Autónomas conheçam e reconheçam a informação utilizada e disponibilizada por cada um dos intervenientes. A única forma de o fazer é recorrendo à normalização.

É com vista na necessidade de integração e de partilha de informação de gestão, que esta nova Arquitectura pretende dar resposta.

Assim, enquanto que a informação de gestão de elementos encontra-se normalizada, quer ao nível da semântica, sintaxe, linguagem e protocolo de comunicação, a informação de gestão de rede ainda não obteve consenso por parte dos construtores ou fabricantes deste tipo de soluções.

Reconhecendo que é uma área complicada, de difícil consenso, é também reconhecida como uma área crucial para que a comunicação entre sistemas gestores heterogéneos possa ser realizável.

4.3 Aplicações de gestão

A importância e interesse na criação de uma estrutura de comunicação com fim à partilha e troca de informação de gestão, surge pelo facto da possibilidade de integração de uma forma transparente entre vários sistemas gestores, de forma a se complementarem, interagirem, ou simplesmente disponibilizarem informação tratada.

A comunicação num ambiente aberto, normalizado e heterogéneo entre Aplicações de Gestão é uma necessidade, que mais tarde ou mais cedo, após permitir uma integração efectiva, irá tender para a delegação e coordenação da gestão, ou seja, a incorporar as funções de gestão de alto nível.

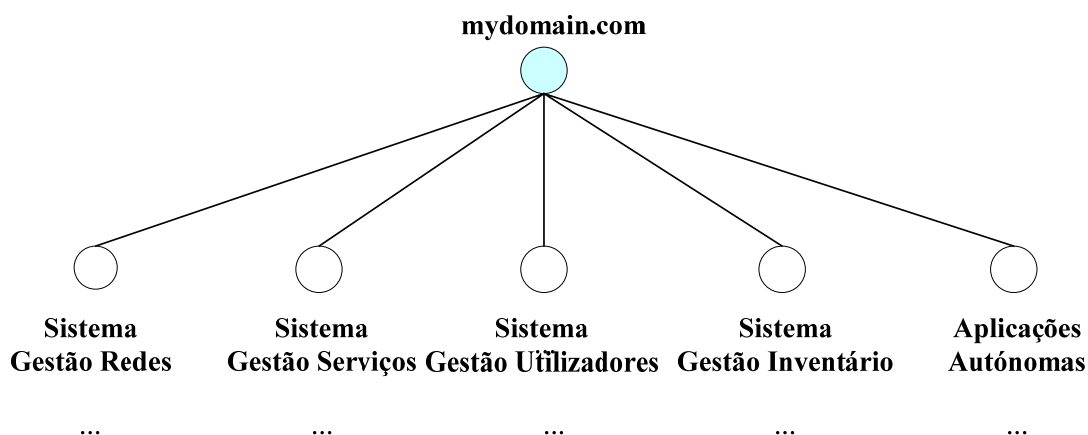


Figura 18 – Árvore de Gestão Simplificada.

A estrutura base para a árvore de informação de gestão de rede tem duas vertentes de utilização:

- **Utilização intra-domínio**, cuja informação é integralmente partilhada e disponível aos sistemas gestores;
- **Utilização inter-domínio**, cuja informação partilhada não diz respeito ao conteúdo do domínio num primeiro nível, mas sim aos sistemas gestores disponíveis e como estes podem ser contactados para funções de gestão inter-domínio.

A informação do Domínio e dos Sistemas Gestores é de carácter público, enquanto a restante informação, e que se encontra dependente dos sistemas gestores é de carácter privado do domínio, pelas razões óbvias.

A informação existente no sistema de directório, a informação de gestão de rede, de serviços, utilizadores e inventário, é fundamentalmente estática ou meta-informação de gestão de redes, daí o reforço na escolha do LDAP.

Esta informação diz respeito a toda a informação de carácter mais permanente que identifica e caracteriza todo ambiente do domínio da rede. São exemplo deste tipo de informação a caracterização dos elementos, a topologia de rede, a relação de serviços, etc. Normalmente, este tipo de informação é de longa duração, a qual é consultada com bastante frequência, mas que está sujeita a poucas alterações, e é assim estruturada e armazenada em sistema de directórios, mais concretamente o LDAP.

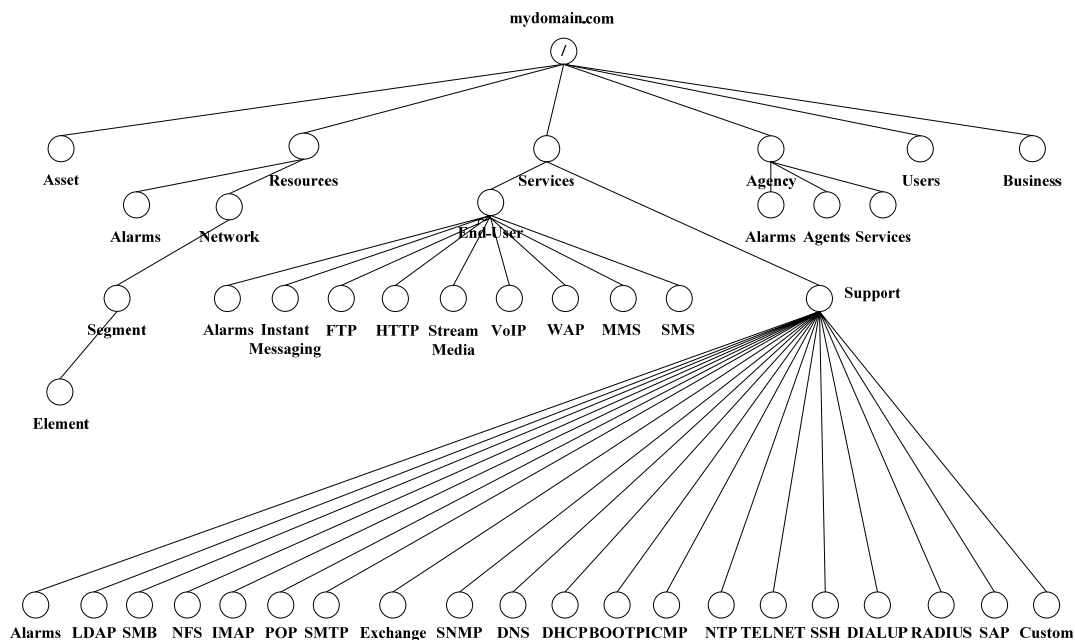


Figura 19 – Árvore de Gestão LDAP Relacionada.

4.3.1 Aplicações Autónomas

De forma a permitir a utilização de Aplicações Autónomas, através do uso do modelo global de informação baseado no LDAP é necessário fazer a sua especificação.

Surge então a necessidade de especificar um *Schema* LDAP de forma a permitir o suporte de todo o ciclo de vida das Aplicações Autónomas.

A árvore de informação (DIT) resultante [26] é baseada na *Abstract Architecture Specification* da FIPA [28], e encontra-se na Figura 20. Aqui, Agentes e Serviços são mantidos separados, uma vez que os agentes podem suportar vários serviços, e um serviço pode ser suportado por um ou mais agentes.

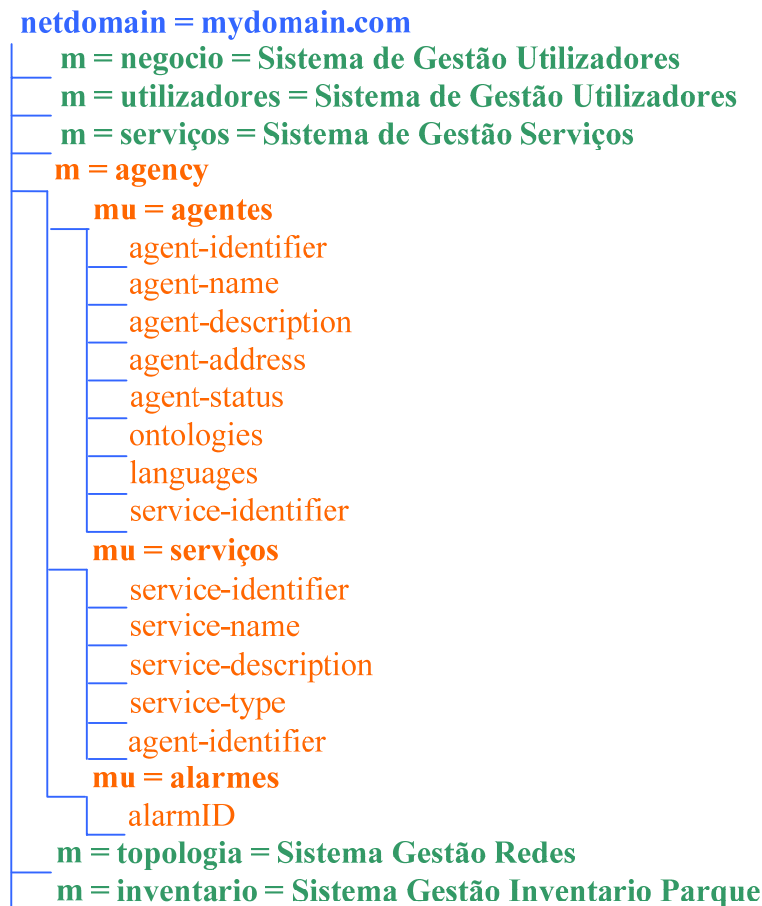


Figura 20 – Árvore de Informação das Aplicações Autónomas.

4.3.2 Directório de Agentes

O Directório de Agentes, é um repositório de informação partilhado, no qual os agentes podem publicar os seus dados, e pesquisar dados de interesse.

De uma forma simples, o papel do Directório de Agentes é fornecer uma localização onde os agentes registam as suas descrições como `agent-directory-entries`. Outros agentes podem pesquisar nas `agent-directory-entries` agentes, com os quais pretendam interagir.

4.3.2.1 agent-identifier

Um `agent-identifier` é um identificador global único de um agente, que pode ser um nome ou número. Vai permitir ao agente ser identificado perante outros agentes.

4.3.2.2 agent-name

Um `agent-name`, é um meio para identificar um agente perante outros agentes e serviços. É expresso num conjunto parâmetro-valor, não é modificável e é único em circunstâncias normais de operação, único e global.

4.3.2.3 agent-description

Contém informação acerca do agente e dos serviços que desempenha.

4.3.2.4 agent-address

Consiste no endereço, que pode ser usado para comunicar com um agente. Um `agent-address` pode ser usado por um serviço de transporte de mensagens para seleccionar um método de transporte para comunicação com o agente, como um agente ou serviço, aqui será endereço de *Instant Messaging*.

4.3.2.5 agent-status

O `agent-status` informação sobre estado do agente, se está ocupado, disponível, ausente, etc. Basicamente reflecte o efeito de presença do *Instant Messaging* [29].

4.3.2.6 Ontologies

Uma ontologia fornece o vocabulário para representação e comunicação de conhecimento sobre um tópico de interesse, e um conjunto de relações e propriedades das entidades existentes nesse vocabulário.

4.3.2.7 Languages

A linguagem usada para expressar o conteúdo da comunicação entre agentes.

4.3.2.8 service-identifier

Um `service-identifier` é um identificador global único do serviço, que permite a identificação do mesmo.

4.3.3 Directório de Serviços

O papel básico do `service-directory-service` é fornecer meios de uma forma consistente, pelos quais agentes e serviços podem descobrir serviços. Em termos operacionais, o `service-directory-service` fornece a localização onde os serviços podem registar as suas descrições como `service-directory-entries`. Agentes e serviços podem pesquisar o `service-directory-service` para localizar serviços apropriados às suas necessidades.

As entradas no `service-directory-service` são descrições de serviços, que consistem em pares parâmetro-valor como `service-name`, `service-type`, `service-locator` e um conjunto opcional de `service-attributes`.

O `service-directory-service` fornece um registo simples para as descrições dos serviços.

Cada `service-directory-service` fornece aos agentes uma `service-root`, que toma a forma de um conjunto de `service-locators`, incluindo pelos menos um `service-directory-service`. De uma maneira geral, o `service-root` vai fornecer entradas suficientes para descrever todos os serviços disponíveis dentro do ambiente, ou vai fornecer apontadores para outros serviços.

4.3.3.1 service-identifier

Um `service-identifier` é um identificador global único do serviço, que permite a identificação do mesmo.

4.3.3.2 service-name

Um nome global e único para o serviço

4.3.3.3 service-description

Contém informação acerca do(s) serviço(s) que desempenha.

4.3.3.4 service-type

É usado para classificar o serviço. O tipo fornece uma referência contextual para a funcionalidade do serviço.

4.3.3.5 agent-identifier

Um `agent-identifier` é um identificador global único de um agente, que pode ser um nome ou número. Vai permitir ao agente ser identificado perante outros agentes.

4.4 Plataforma de Comunicação

A comunicação neste modelo entre Aplicações Autónomas e Utilizadores vai ser baseada em *Instant Messaging*. Aqui o Serviço de Transporte de Mensagens [8] vai dar lugar ao *Instant Messaging*, através do uso do protocolo XMPP [30], o que possibilita o transporte de conteúdo XML.

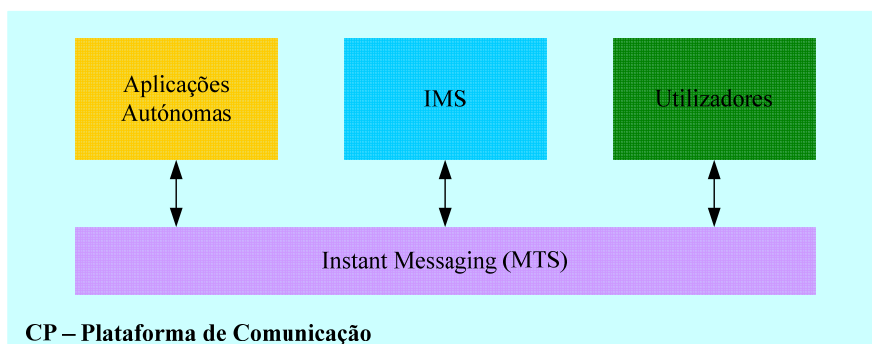


Figura 21 – Plataforma de Comunicação.

Na figura anterior é possível ver a Plataforma de Comunicações para o modelo proposto por este trabalho. Aqui as Aplicações Autónomas e os Utilizadores realizam o registo no servidor de *Instant Messaging*, de forma a ser possível a comunicação com este Serviço de Transporte de Mensagens. De notar também que o servidor de *Instant Messaging* irá realizar a autenticação no servidor de LDAP usado como modelo global de informação de forma a termos uma maior integração dos Sistemas descritos ao longo deste trabalho.

Esta plataforma fornece a capacidade de Utilizadores e Aplicações Autónomas comunicarem entre si, bem como saber qual o seu estado em determinado instante, através do uso do *Instant Messaging* e do efeito presença.

O efeito presença descreve o estado de uma entidade, que pode ser uma Aplicação Autónoma ou Utilizador, num determinado instante, como por exemplo, disponível, não disponível, ocupado, etc. Desta forma é assim possível para um Utilizador saber o estado de uma Aplicação Autónoma.

A comunicação é normalizada, comum e consequentemente mais simples, quer isto dizer que a comunicação é feita através de *Instant Messaging*, mais concretamente o protocolo XMPP com recurso ao XML como suporte à linguagem.

Para as Aplicações Autónomas, o uso de XML é suficiente uma vez que eles não precisam de interface gráfica, enquanto que os Utilizadores Humanos (Administradores ou Operadores de rede) precisam. Facilmente se pode recorrer às folhas de estilo XSL combinadas com o XML, para a criar uma representação mais amigável do ponto de vista do Utilizador Humano.

4.4.1 Protocolos Comunicação

Neste modelo proposto, existem pelo menos três níveis de comunicação a serem considerados:

- Comunicação entre as Aplicações Autónomas e o Barramento de Informação de Gestão LDAP;
- Comunicação entre Aplicações Autónomas e entre Aplicações Autónomas e Utilizadores;
- Comunicação entre Utilizadores e o Barramento de Informação de Gestão LDAP.

Quanto ao primeiro nível, o protocolo LDAP assume as funções de responder às necessidades das Aplicações Autónomas em aceder ao modelo global de informação baseado no LDAP (Figura 22).

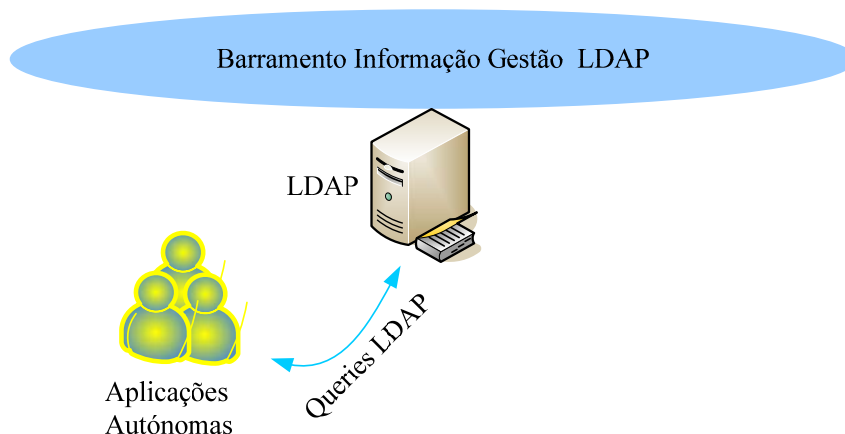


Figura 22 – Comunicação entre Aplicações Autónomas e LMIB.

No que diz respeito ao segundo nível da comunicação, entre Aplicações Autónomas e entre Aplicações Autónomas e Utilizadores, esta é assegurada pelo *Instant Messaging*, mais concretamente o protocolo XMPP uma vez que tem suporte para o transporte de conteúdo XML (Figura 23).

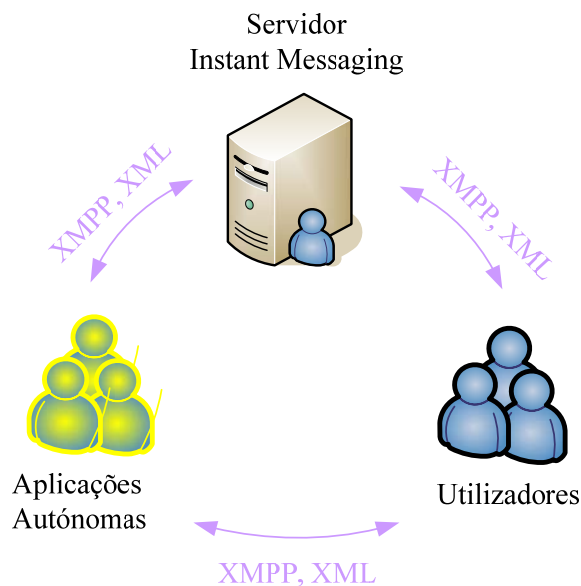


Figura 23 – Comunicação entre Aplicações Autónomas e Utilizadores.

Finalmente no terceiro e último nível de comunicação, entre Utilizadores e o Barramento de Informação de Gestão LDAP, a comunicação é feita através da implementação de uma Consola de Gestão. Esta consola faz uso das linguagens de conteúdo XML e XSL, e que por sua vez são transportadas através do uso do protocolo de transporte HTTP para comunicação com o Utilizador.

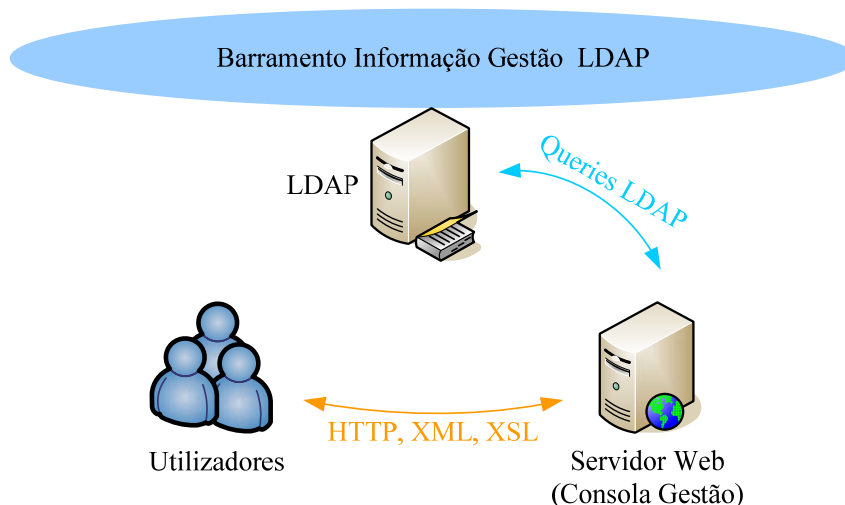


Figura 24 – Comunicação entre Utilizadores e LMIB.

Neste modelo, a utilização de protocolos normalizados e a aceitação do ambiente de funcionamento Internet como uma realidade incontornável, faz com que o recurso aos protocolos já existentes ao nível da camada de aplicação seja uma mais valia para a Arquitectura proposta. Agora, a comunicação que antes era só analisada na perspectiva Aplicação de Gestão – Utilizadores, assume aqui um novo papel, que é atribuído pela utilização da consola e do modelo de informação global LDAP de uma forma autónoma e modularizada.

Dado não existir ainda nenhuma normalização quanto à comunicação de gestão, a este nível, optou-se pela utilização dos protocolos normalizados e mais utilizados no âmbito da Internet. Neste modelo, a comunicação em ambiente de gestão é equivalente à das outras actividades, utiliza o mesmo meio, pelo que os protocolos existentes podem perfeitamente servir de base à actividade de comunicação de gestão entre sistemas gestores e entre estes e as consolas de gestão.

A nova Arquitectura de Comunicação, pretende alterar a arquitectura dos Sistemas de Gestão actuais, de forma a manter o que é bom nos sistemas actuais, comunicação entre agente – sistema gestor e os serviços de gestão, e por outro lado, normalizar e autonomizar o sistema de informação de gestão de redes que vai além do Objecto Gerido, tratado pelo SNMP e com o recurso à MIB, bem como criar um nível de comunicação de gestão entre sistemas gestores e entre estes e as consolas.

4.4.2 Protocolos de Gestão

Os protocolos de Gestão utilizados nesta Arquitectura são os protocolos já abordados no Capítulo anterior, ou seja, o SNMP e o CMIP.

Os protocolos eleitos para a gestão são assim o SNMP e CMIP. Estes protocolos vão actuar sobre os elementos de rede, *hubs*, *switchs*, *routers*, etc. de forma a solucionarem os problemas existentes.

4.4.3 LDAP

O protocolo LDAP assume aqui a responsabilidade de implementar o Barramento de Informação de Gestão LDAP, bem como responder às necessidades das Aplicações Autónomas, como por exemplo na pesquisa de agentes ou serviços.

Conforme dito anteriormente, ele é usado na comunicação entre Aplicações Autónomas e o Barramento de Informação de Gestão LDAP.

É necessário que Aplicações Autónomas estejam registadas no servidor de LDAP. A seguir, na Tabela 7, é apresentado um código exemplo, para o registo de Aplicações Autónomas no LDAP.

Tabela 7 – Exemplo de registo no LDAP.

```
# Registo de Aplicação Autónoma na LDAP
# Agent Account [dispatcher]
dn: uid=dispatcher,ou=Agentes,ou=Agency,dc=mydomain,dc=com
cn: Dispatcher Agent
objectClass: top
objectClass: agent
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
uid: dispatcher
```

```
userPassword: password  
structuralObjectClass: inetOrgPerson  
mail: dispatcher@mydomain.com
```

4.4.4 Instant Messaging

O *Instant Messaging* envolve comunicações electrónicas síncronas onde dois ou os mais utilizadores trocam mensagens de texto ou voz através de dispositivos electrónicos.

Os sistemas primitivos de troca de mensagens tornaram-se disponíveis há muitos anos, como por exemplo, o UNIX *Talk* em 1973 e o *Internet Relay Chat* em 1988. Do ponto de vista dos utilizadores finais, estes sistemas operavam de forma similar ao telefone, na medida em que o utilizador fazia uma “chamada cega”, na esperança de que o receptor estivesse disponível para atender a chamada de forma a completar a mesma.

A estes sistemas faltava a informação de presença associada. O *Instant Messaging* na sua forma actual foi disponibilizado pela rede ICQ em 1996. Como consequência, várias companhias como a AOL e a Microsoft estabeleceram redes públicas de consumidores, baseadas em vários protocolos similares mas incompatíveis, enquanto que empresas como a IBM se concentraram em fornecer sistemas de mensagens corporativas privadas.

O IETF e a fundação do software de *Jabber* [30] investiram recursos consideráveis no desenvolvimento de propostas de protocolos tais como o SIP *Instant Messaging Presence Leveraging Extensions* (SIMPLE) e o *Extensible Messaging and Presence Protocol* (XMPP), cada um concentrado em fornecer uma norma para *Instant Messaging* em redes fixas.

4.4.4.1 Características dos Sistemas *Instant Messaging*

Na sua forma mais simples, o *Instant Messaging* combina comunicações bidireccionais juntamente com informação que descreve o estado de cada utilizador na rede. Uma conversa em forma de texto pode ser formada através da troca das mensagens curtas e simples que são entregues em aproximadamente tempo real.

O aspecto mais importante do IM actual, que o diferencia dos sistemas antigos, é a integração do efeito presença, fornecendo assim a habilidade de monitorar o estado de outros utilizadores na rede.

Na maioria das aplicações cliente de IM, esta informação é apresentada na forma de uma lista de contactos, que indica o estado actual dos mesmos.

Se um utilizador em particular estiver *online*, um indicador vai mostrar esta informação de presença nas listas de contactos daqueles utilizadores que subscreveram a informação de presença do utilizador em causa.

É esta combinação do estado dos utilizadores e da comunicação quase em tempo real que fornece o efeito de estado e presença, agindo como um pivot na adopção deste método de comunicação.

4.4.4.2 Presença e disponibilidade

A informação de presença, refere-se a dados capazes de descrever o estado de um utilizador e a disponibilidade de receber várias formas de comunicação. Por exemplo, esta informação pode descrever se um utilizador é capaz de receber uma chamada telefónica, ou uma mensagem instantânea num dado momento. A informação de presença também pode ser estendida para suportar informação complementar como a informação geográfica, etc.

A disponibilidade denota a “vontade” de um utilizador em comunicar, dependendo de um contexto específico, geralmente determinado pelo tipo de comunicação que é pedida e pela identidade do utilizador que faz o pedido.

Por exemplo, enquanto um utilizador de *Instant Messaging* for capaz de receber mensagens de qualquer outro utilizador na mesma rede, na prática ele pode querer estar indisponível para toda a gente, excepto para os utilizadores da sua lista de contactos. Esta situação evita mensagens não solicitadas.

Em sistemas de *Instant Messaging*, os conceitos de presença e disponibilidade geralmente estão ligados um ao outro.

Uma aplicação cliente típica, informa o utilizador quando o estado de um contacto é alterado, e fornece também a capacidade de alterar o seu estado facilmente.

O *Instant Messaging* tem nesta Arquitectura um papel fundamental, na medida em que é usado na comunicação entre Aplicações Autónomas, para trocarm informação de gestão, e entre Aplicações Autónomas e Utilizadores, não só para troca de informação de gestão, mas também para os Utilizadores saberem qual o estado das Aplicações Autónomas ou então para delegarem tarefas.

4.5 Linguagens Comunicação

4.5.1 Ontologias

Uma ontologia é uma especificação ou formalização de uma conceptualização [31]. Alternativamente, uma ontologia é uma teoria lógica, a qual dá uma explicação de uma conceptualização, projectada para ser partilhada por Agentes com vários objectivos [32]. Uma conceptualização é descrita por um conjunto de conceitos e pelas relações entre si.

Uma ontologia também pode ser vista como uma forma de descrever um entendimento comum e partilhável, sobre o tipo de objectos e de relacionamentos que estão a ser descritos, tal que a comunicação possa acontecer entre pessoas e sistemas [33]. Noutras palavras, é a terminologia de um domínio (define o universo de discurso). Como um exemplo real, pode-se considerar um *thesaurus* usado para a procura uniforme sobre um conjunto de páginas *Web* semelhantes, porém independentes.

As ontologias podem ser usadas para:

- Criar um vocabulário central estruturado, para ser usado e validado por um conjunto de actores numa comunidade;

- Definir e usar relacionamentos lógicos e regras entre os conceitos, permitindo um uso eficiente de Aplicações Autónomas;
- Desenvolver, manter e publicar conhecimento (que pode sofrer mudanças rapidamente) sobre uma organização, rede informática (toda ou parte dela), provendo facilmente visões distintas sobre a mesma.

Na Arquitectura de Comunicação proposta é também necessário o desenvolvimento de uma ontologia, uma vez que é necessário que os Utilizadores e as Aplicações Autónomas disponham de uma forma de entendimento comum. Se a ontologia não for criada, corre-se o risco de as mensagens trocas não serem compreendidas pelos intervenientes.

Uma vez que estamos a falar da criação de ontologias para a Gestão de Redes IP, tem de se ter em atenção sobre os vários níveis de gestão existentes. Esta situação é particularmente interessante e importante, na medida em que se trata de uma Arquitectura Multi-Agente. Não faz sentido criar uma só ontologia para este sistema, faz sentido sim, relacionar os níveis de gestão de redes com as áreas funcionais.

Conforme se pode ver na Figura 25, é possível ter uma ontologia para configuração. O problema é que a configuração existe em todos os Níveis da Gestão. É possível criar uma ontologia mais específica e não generalista, por exemplo uma ontologia para configuração de Serviços, outra para Utilizadores, etc.

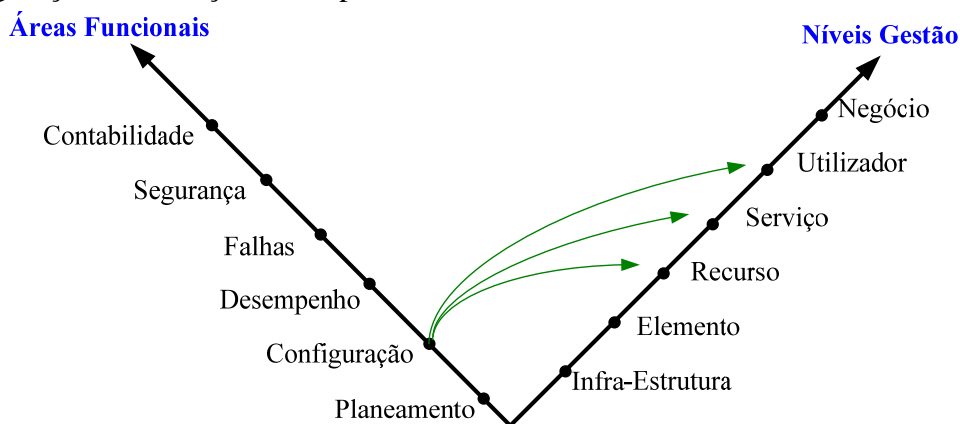


Figura 25 – Ontologia com relação Área funcional com Níveis da Gestão.

Desta forma é possível ter uma forma de entendimento comum e partilhável entre Aplicações Autónomas e entre estas e os Utilizadores. Assim é possível ter ontologias específicas a cada uma das Áreas Funcionais, que serão executadas por Aplicações Autónomas, relacionadas com o nível de gestão em causa.

4.5.2 XML

A linguagem XML, como subconjunto da linguagem SGML, é usada para a representação de dados. Por exemplo a informação de gestão em formato texto. Uma das vantagens do XML prende-se com a possibilidade de, opcionalmente, incorporar-se a própria gramática em anexo, permitindo assim enviar dados e a estrutura dos dados utilizada.

A gramática para um documento XML é descrita através do mecanismo conhecido como DTD que é responsável por descrever os elementos num documento XML. Para que a informação de gestão possa utilizar convenientemente o XML é necessário definir um vocabulário e DTDs.

Os documentos XML contêm informação, cuja forma de captar ou de a mostrar pode ser alcançada com o uso de documentos XSL.

Assim, um documento XML associado a um XSL pode ser utilizado para a apresentação e representação gráfica da informação contida no documento.

O XML é usado na comunicação de informação em dois níveis diferentes:

- Na comunicação de informação entre Aplicações Autónomas e entre Aplicações Autónomas e Utilizadores;
- Na comunicação entre Utilizadores e o Barramento de Informação de Gestão LDAP, via Consola de Gestão.

Quanto ao primeiro nível, a utilização do XML deve-se ao facto da utilização do protocolo de *Instant Messaging* XMPP, e também devido à linguagem de comunicação entre agentes FIPA-ACL com representação em XML [34].

No que se refere ao segundo nível, a utilização do HTML, do XML e XSL como linguagens base na comunicação entre os Utilizadores e as Consolas de Gestão é algo que tem vindo a ser adoptado.

4.5.3 Mensagens

Com base nos protocolos de comunicação abordados mais atrás, vamos ver alguns exemplos de mensagens trocadas em cada um dos níveis já mencionados, ou seja:

- **Queries LDAP:** Usado em comunicações entre as Aplicações Autónomas e o Barramento de Informação de Gestão LDAP;
- **Instant Messaging:** Usado na comunicação entre Aplicações Autónomas e entre Aplicações Autónomas e Utilizadores;

4.5.3.1 Queries LDAP

As *queries* LDAP, conforme dito anteriormente são usadas para comunicação entre Aplicações Autónomas e o Barramento de Informação de Gestão LDAP, por exemplo para pesquisa de Agentes ou Serviços.

Para fazer pesquisas no servidor de LDAP, é necessário saber o que se quer pesquisar e onde. Sendo assim na *query* LDAP a utilizar, é necessário indicar em que ramo da Árvore de Informação se quer pesquisar.

Sendo assim, o ramo da Árvore de Informação a pesquisar é onde estão as Aplicações Autónomas, ou seja, em `ou=agentes,ou=Agency,dc=mydomain,dc=com`. Neste caso em concreto, é possível pesquisar por qualquer atributo como o `agent-identifier`, `agent-name`, `service-name`, `agent-description`, `service-description`, etc.

Tabela 8 – Exemplo de Pesquisa e Resultado no LDAP.

```
#Pesquisa no LDAP

ldap_search_s(ld, "ou=agentes,ou=agency,dc=mydomain,dc=com", 2,
(&(objectClass=agent)(service-name=SGU)))" ,attrList,0,&msg)

# Resultado da Pesquisa

dn: cn=SGU,ou=agentes,ou=agency,dc=mydomain,dc=com
cn: SGU;
displayname: Agente SGU;
mail: sgu@mydomain.com
description: Agente Sistema Gestão Utilizadores;
```

Esta mensagem tem como objectivo pesquisar um agente/serviço no servidor LDAP.

4.5.3.2 Envio de mensagens *Instant Messaging*

As mensagens instantâneas, são usadas para comunicação entre Aplicações Autónomas e entre estas e Utilizadores.

Para enviar uma mensagem instantânea é necessário, à semelhança de um e-mail, indicar qual o destinatário da mensagem, o remetente, o assunto da mensagem e claro a mensagem em si.

Na Tabela 9 é possível ver um exemplo de uma mensagem *Instant Messaging*, em que os campos necessários estão a negrito.

Tabela 9 – Exemplo de Mensagem IM.

```
#DispatcherAgent envia dados ao agente responsável

<?xml version="1.0"?>

<stream:stream xmlns:stream="http://etherx.jabber.org/streams"
xmlns="jabber:client" to="mydomain.com">

<message to=destino@mydomain.com from=origem@mydomain.com >
<subject>Texto_do_Assunto</subject>
<body>
Corpo da mensagem.
</body>
</message>
</stream:stream>
```

Esta mensagem tem como objectivo enviar uma mensagem de origem@mydomain.com para destino@mydomain.com, com o Assunto “*Texto_do_Assunto*”, e como corpo da mensagem “*Corpo da mensagem*”.

4.5.3.3 Conteúdo das mensagens IM

Uma mensagem sem conteúdo não faz muito sentido. É como receber um envelope vazio. Sendo assim torna-se necessário usar uma linguagem de conteúdo, neste caso o FIPA-ACL com referência ao XML.

Nesta Arquitectura, a mensagem de conteúdo é constituída por vários campos entre os quais:

- **Performativa:** Indica qual a acção tomar sobre a mensagem;
- **Linguagem:** Usado na comunicação entre Aplicações Autónomas e entre Aplicações Autónomas e Utilizadores, indica qual a linguagem de comunicação de agentes usada;
- **Ontologia:** Usado para indicar qual a ontologia usada, de forma a permitir o entendimento comum sobre o tipo de objectos e de relacionamentos que estão a ser descritos;
- **Conteúdo**
 - No conteúdo existe ainda o tipo de alerta, que pode ser *Error*, *Alarm*, *Notify*.
 - **Serviço afectado:** Indica qual o serviço afectado;
 - **Descrição:** contém uma descrição sobre o alarme, mais concretamente sobre o serviço afectado;
 - **Número do ticket:** Identificação do número do *Trouble Ticket*.

Na Tabela 10 é possível ver uma Mensagem de Conteúdo.

Tabela 10 – Exemplo de Mensagem de Conteúdo.

```
# Conteúdo da Mensagem
<Inform>
<language>fipa-s10</language>
<ontology>AgentsManagement</ontology>
<content>
  <error>Sistema de Gestão de Serviços</error>
  <service>DNS</service>
  <description>Cannot resolve name.</description>
  <ticket>1</ticket>
</content>
</Inform>
```

Esta mensagem de conteúdo, utiliza a ontologia *AgentManagement*, a linguagem *fipa-s10*, relata um erro no Sistema de Gestão de Serviços, em que o serviço afectado por esse erro é o DNS. A descrição do problema é que não é possível traduzir nomes, e o pedido tem o *ticket* número um.

O conteúdo das mensagens a enviar por *Instant Messaging*, pode ter o próprio conteúdo diferente, devido às necessidades de comunicação, por exemplo, devido às áreas funcionais de gestão.

4.6 Segurança

Actualmente, existem diversas questões de segurança associadas aos Sistemas de *Instant Messaging*, como:

4.6.1 Vírus

Os Sistemas de *Instant Messaging*, são uma “ferramenta” baseada no uso da Internet, como tal, sofrem de vulnerabilidades comuns a outras aplicações baseadas no uso da Internet. Até à data, existem mais de 200 vírus reportados, relacionados com este tipo de sistemas. [35]

O efeito de ataques de vírus nestes sistemas, é mais crítico no *Instant Messaging*, quando comparados aos ataques baseados nos sistemas de E-mail, uma vez que os ataques podem espalhar-se muito mais rapidamente. Por outro lado, os administradores dos sistemas *Instant Messaging*, necessitam de reagir muito mais rapidamente, de forma a controlar a propagação dos vírus.

4.6.2 Mensagens de Spim

As mensagens de SPAM, no contexto do *Instant Messaging*, são mencionadas por mensagens SPIM.

As mensagens não desejadas de SPIM, podem aparecer (fazer *pop-up*) em qualquer altura, e desta forma interromper o trabalho momentaneamente.

De forma análoga ao SPAM, existem ferramentas Anti SPIM disponíveis para bloquear a recepção deste tipo de mensagens, e permitir que Utilizadores e Aplicações Autónomas não sejam afectadas com este tipo de mensagens.

4.6.3 Privacidade

A informação de privacidade é um caso delicado, na medida em que a informação relativa a Utilizadores e Aplicações Autónomas, não deve ser passada ou distribuída, sem a permissão dos mesmos a terceiros.

A informação de presença, se enviada sem permissões, por exemplo a agências promocionais, pode prejudicar os interesses dos Utilizadores.

4.7 Conclusão

Este capítulo apresentou a Arquitectura de Comunicação do Sistema de Aplicações Autónomas de Gestão

Esta Arquitectura tenta alcançar a integração e partilha da informação de gestão, através do Modelo Global de Informação baseado Árvores de Informação ou Sistemas

de Directório, bem como utilizar o uso de *Instant Messaging* como serviço de Transporte de Mensagens.

Inicialmente foi feita uma descrição das Árvores de Informação utilizadas e dos seus aspectos funcionais. Foram também descritos os protocolos de comunicação e gestão, bem como as linguagens utilizadas.

Por fim, foram abordadas algumas questões de segurança, mais concretamente a segurança nos Sistemas de *Instant Messaging*.

Capítulo 5

5 Protótipo

Um problema que é possível antever é o que acontece se o gestor humano não está fisicamente no seu domínio de gestão. Como é possível executar tarefas de gestão?

O cenário operacional aqui apresentado é semelhante a uma casa a arder, em que o dono recebe um e-mail com fotografias ou vídeos da sua casa em chamas. Uma vez que ele não se encontra no local, ele não pode fazer nada a não ser ver a sua casa a arder! Devido a este tipo de problemas, este conceito fornece um meio para actuar de forma rápida, quase em tempo real, através do uso de Aplicações Autónomas e de *Instant Messaging*, que agem em nome do gestor humano.

Neste contexto em que se tem vindo a apresentar um novo conceito de gestão, é objectivo deste capítulo apresentar um protótipo de utilização para a gestão de redes, através do uso da nova arquitectura de Gestão de Redes juntamente com Aplicações Autónomas.

5.1 Ambiente de Gestão Intra-Domínio

O nosso ambiente de gestão de redes intra-domínio é composto pela Intranet, que é a rede local onde é possível encontrar os postos de trabalho e os vários de serviços de rede disponibilizados, como por exemplo o correio electrónico, o *Instant Messaging*, entre outros, e a DMZ que é uma sigla para “*DeMilitarized Zone*” ou zona desmilitarizada.

Também conhecida como Rede de Perímetro, a DMZ é uma pequena rede situada entre uma rede confiável e uma não confiável, geralmente entre a rede local e a Internet.

A função de uma DMZ é manter todos os serviços que possuem acesso externo (HTTP, DNS, *Mail Relay*, etc.) separados da rede local, limitando o dano em caso de comprometimento de algum serviço nela presente por algum invasor. O termo possui uma origem militar, significando a área existente entre dois inimigos numa guerra.

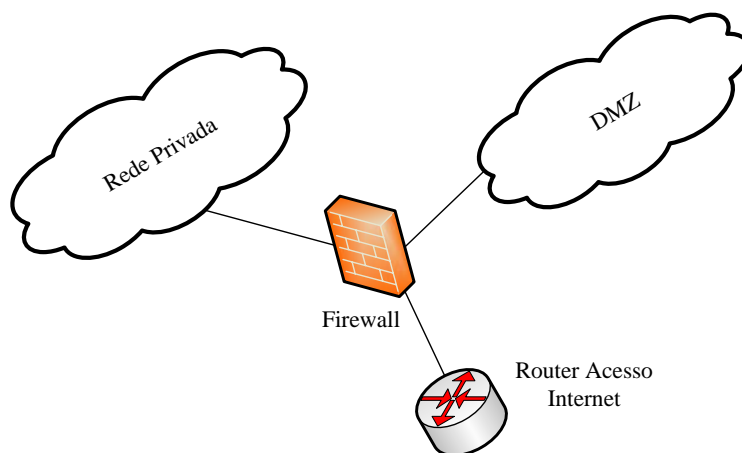


Figura 26 – Ambiente de rede Intra-Domínio.

No ambiente de gestão de redes intra-domínio, é possível encontrar várias aplicações de gestão nomeadamente um Sistema de Gestão de Redes, responsável por gerir a infraestrutura de TI, um Sistema de Gestão de Serviços, responsável por gerir os serviços existentes na rede, um Sistema de Gestão de Utilizadores, usado para gerir perfis de utilizadores, e finalmente um Sistema de Gestão de Inventário do Parque responsável pelo inventário da rede.

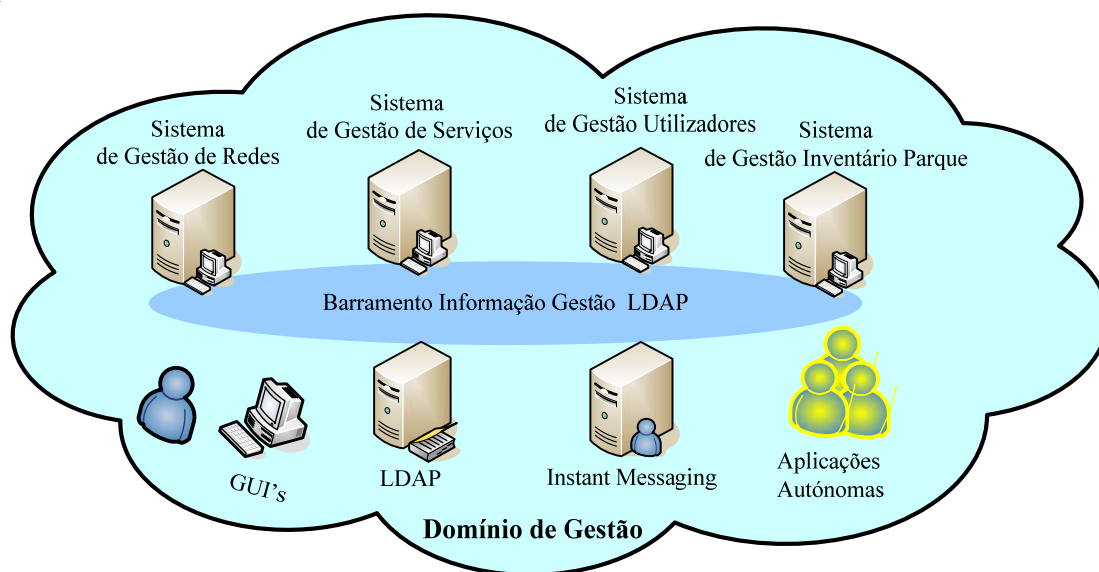


Figura 27 – Ambiente de Gestão.

Assumindo assumir que as várias aplicações de gestão já têm implementado uma interface LDAP, conforme abordado no capítulo 3, e que toda a informação de gestão necessária está no modelo global de informação baseado no LDAP.

Além das aplicações de Gestão referidas acima, é possível encontrar também outros servidores no nosso ambiente de gestão, como é o caso de servidores de *Instant Messaging*, servidores de Correio Electrónico (IMAP, SMTP, POP3), servidores *Web*, servidores LDAP, entre outros.

O servidor LDAP, responsável por implementar o barramento de Informação de Gestão (LMIB), suporta o modelo global de informação de gestão da rede, de forma a tirar partido da existência de interfaces LDAP nas aplicações de gestão descritas anteriormente.

Também é possível encontrar um servidor de *Instant Messaging* (XMPP/XML), responsável pelo transporte de mensagens entre administradores ou operadores de rede e entre Aplicações Autónomas.

Os administradores ou operadores de rede, têm estações de trabalho a disponibilizarem a consola de gestão. Através das consolas de gestão, os administradores ou operadores de rede podem executar tarefas de gestão através de protocolos normalizados de Internet (HTTP, XML/XSL) sem recorrerem a interfaces de gestão ou APIs proprietárias.

Por outro lado o uso de Aplicações Autónomas vai permitir automatizar algumas tarefas de gestão de redes, uma vez que a informação de gestão está no modelo global de informação baseado no LDAP, e que é acessível de forma aberta e normalizada através do servidor de LDAP.

Considerando que a forma de aceder à informação de gestão é aberta e normalizada, e que a informação de gestão das várias aplicações de gestão reside no modelo global de informação baseado em LDAP, a comunidade de Aplicações Autónomas existente pode facilmente aceder à informação de gestão dos vários níveis (Físico, Elemento, Recurso, Utilizador, Negócio), sempre da mesma forma, bem como fazer/aceder às relações entre eles.

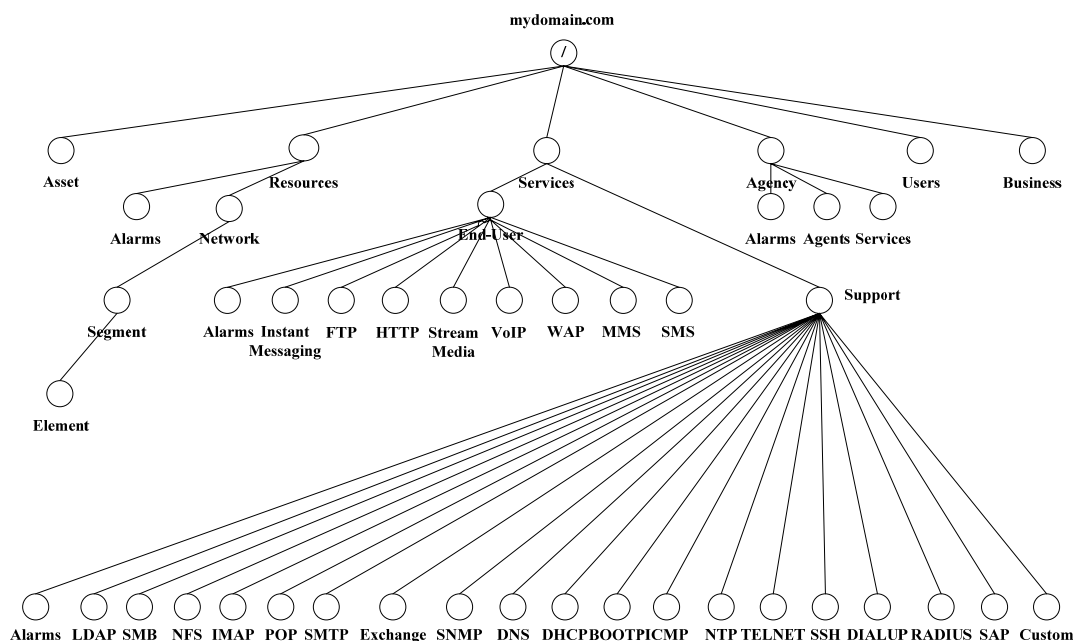


Figura 28 – Árvore de Informação de Gestão.

A seguir é possível ver a integração das aplicações de gestão, utilizadores e Aplicações Autónomas recorrendo a protocolos de Internet que toda a gente conhece e usa.

5.1.1 Protótipo de utilização autónomo

Neste protótipo vamos assumir que um utilizador está a ter problemas com o serviço de correio electrónico. O utilizador contacta o serviço de *Helpdesk* a reportar o problema e é aberto um *ticket* no Sistema de Gestão de Utilizadores.

Aqui as aplicações de gestão realizam as tarefas que lhes competem. O Sistema de Gestão de Redes, gere a infra-estrutura, o Sistema de Gestão de Serviços, gere os serviços de rede, o Sistema de Gestão de Utilizadores, gere perfis de utilizadores, e finalmente o Sistema de Gestão de Inventário do Parque faz o inventário da rede.

A informação de gestão proveniente destas aplicações é guardada no modelo global de informação baseado no LDAP, através do uso de interfaces LDAP das aplicações de gestão, que devem de estar implementadas nas mesmas.

Devido ao *ticket* existente no Sistema de Gestão de Utilizadores, este sistema gera um alarme para resolução do problema, e através de uma interface IMAP envia um e-mail para o administrador ou operador de rede com os dados do alarme.

Uma Aplicação Autónoma chamada de *Dispatcher Agent*, é responsável por receber os alarmes gerados pelas aplicações de gestão (Figura 29 (1)), uma vez que tem mapeada uma conta IMAP nas aplicações de gestão existentes. Além de receber os alarmes, também é responsável pela sua análise. Isto significa que o *Dispatcher Agent* pode fazer pesquisas no servidor de LDAP (Figura 29 (2)) de forma a encontrar um agente/serviço capaz de resolver o problema, por exemplo através do nível de gestão a que o alarme pertence.

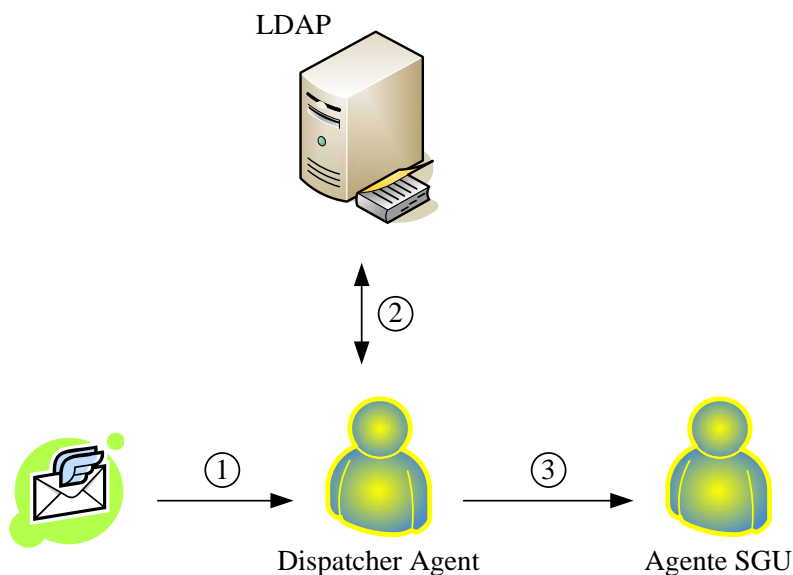


Figura 29 – Protótipo Inter Domínio 1.

Uma vez que o *Dispatcher Agent* precisa saber qual o servidor de LDAP a contactar, é necessário criar um registo SRV no DNS para o serviço LDAP. Este registo descreve a máquina a ser contactada para acesso à informação de gestão de rede, tal como é utilizado hoje em dia o registo MX para os servidores de correio electrónico.

Tabela 11 – Pesquisa/Resultado por Agente/Serviço de Utilizadores no LDAP.

#Pesquisa no LDAP

```
ldap_search_s(ld, "ou=Agentes,ou=agency,dc=mydomain,dc=com", 2,
(&(objectClass=agent)(service-name=SGU)))" ,attrList,0,&msg)
```

Resultado da Pesquisa

```
dn: cn=SGU,ou=Agentes,ou=agency,dc=mydomain,dc=com
cn: SGU;
displayname: Agente SGU;
mail: sgu@mydomain.com
description: Agente Sistema Gestão Utilizadores;
```

Assim que o agente/serviço é encontrado, o *Dispatcher Agent* vai iniciar uma sessão de *Instant Messaging* com o agente SGU, que conforme o seu estado de presença, vai passar a informação do alarme via (Figura 29 (3)), de forma a dar seguimento à resolução do problema.

Tabela 12 – Envio do alarme via *Instant Messaging* pelo *Dispatcher Agent*.

#DispatcherAgent envia dados ao agente responsável

```
<?xml version="1.0"?>
<stream:stream xmlns:stream="http://etherx.jabber.org/streams"
xmlns="jabber:client" to="mydomain.com">
<message to=sgu@mydomain.com from=dispatcher@mydomain.com >
<subject>Alarm Message!!!</subject>
<body>
  <Inform>
    <language>fipa-s10</language>
    <ontology>AgentsManagement</ontology>
    <content>
      <error>Sistema de Gestão de Utilizadores</error>
      <service>Correio Electrónico</service>
      <description>Não recebe mensagens de e-mail.</description>
      <ticket>16</ticket>
    </content>
  </Inform>
</body>
</message>
</stream:stream>
```

O agente SGU recebe os dados do alarme por *Instant Messaging* e após fazer a sua análise, verifica que um utilizador se queixa do serviço de Correio Electrónico. Uma

vez que o agente SGU não é responsável pelo serviço de Correio Electrónico, ele vai fazer uma pesquisa no LDAP por um agente responsável pelo mesmo (Figura 30 (1)).

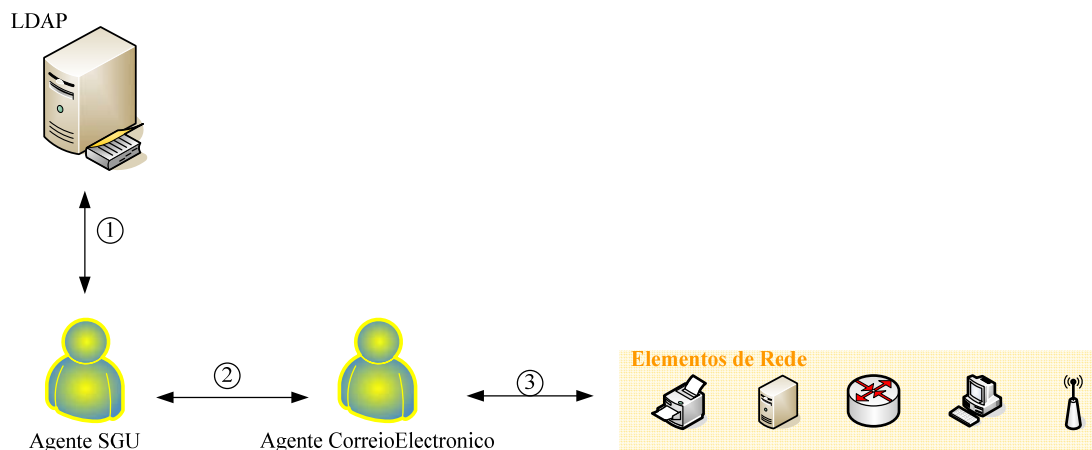


Figura 30 – Protótipo Inter Domínio 2.

Tabela 13 – Pesquisa/Resultado por Agente/Serviço de Correio Electrónico no LDAP.

```
#Pesquisa no LDAP
ldap_search_s(ld, "ou=Agentes,ou=agency,dc=mydomain,dc=com", 2,
(&(objectClass=agent)(service-name=CorreioElectronico)))", attrList,0,&msg)

# Resultado da Pesquisa
# Resultado da Pesquisa
dn: cn=CorreioElectronico,ou=Agentes,ou=agency,dc=mydomain,dc=com
cn: CorreioElectronico;
displayname: Agente CorreioElectronico;
mail: correioelectronico@mydomain.com
description: Agente Correio Electrónico;
```

Após o agente ser encontrado o agente SGU encaminha o alarme para o agente de Correio Electrónico via *Instant Messaging* (Figura 30 (2)).

Tabela 14 – Encaminhamento do alarme pelo agente SGU para o agente CorreioElectronico.

```
#Agente SGU envia dados ao agente CorreioElectronico
<?xml version="1.0"?>
<stream:stream xmlns:stream="http://etherx.jabber.org/streams"
xmlns="jabber:client" to="mydomain.com">
<message to=correioelectronico@mydomain.com from=sgu@mydomain.com >
<subject>Alarm Message!!!</subject>
<body>
    <Inform>
    <language>fipa-s10</language>
```

```

<ontology>AgentsManagement</ontology>

<content>

    <error>Sistema de Gestão de Utilizadores</error>

    <service>Correio Electrónico</service>

    <description>Não recebe mensagens de e-mail.</description>

    <ticket>16</ticket>

    <notify>sgu@mydomain.com</notify>

</content>

</Inform>

</body>

</message>

</stream:stream>

```

Após receber alarme, o agente CorreioElectronico vai consultar no LDAP a árvore de gestão para saber qual a máquina que serve o Correio Electrónico. Depois de descobrir qual é a máquina, o agente CorreioElectronico vai fazer um conjunto de procedimentos, previamente programados de forma a saber qual o problema. Ele começa por iniciar uma sessão remota ao servidor de Correio Electrónico para consultar os *logs* que foram gerados (Figura 30 (3)). Ao consultar os *logs* ele verifica que existem várias linhas em que o servidor tinha registado um erro “...cannot resolve name...”, que é um erro típico de DNS.

Uma vez que o agente CorreioElectronico não é responsável pelo serviço de DNS, ele vai pesquisar no LDAP por um agente responsável pelo mesmo e vai reencaminhar o alarme (Figura 31 (1) (2)).

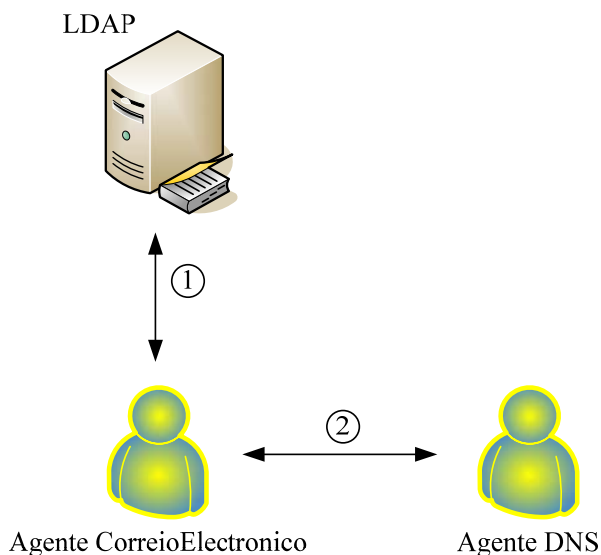


Figura 31 – Protótipo Inter Domínio 3.

Tabela 15 – Pesquisa/Resultado por Agente/Serviço de DNS.

#Pesquisa no LDAP

```
ldap_search_s(ld, "ou=Agentes,ou=agency,dc=mydomain,dc=com", 2,
(&(objectClass=agent)(service-name=DNS)))" ,attrList,0,&msg)

# Resultado da Pesquisa
dn: cn=DNS,ou=Agentes,ou=agency,dc=mydomain,dc=com
cn: DNS;
displayname: Agente DNS;
mail: dns@mydomain.com
description: Agente Domain Name Service;
```

Tabela 16 – Encaminhamento do alarme pelo agente CorreioElectronico para o agente DNS.

```
#Agente CorreioElectronico envia dados ao agente DNS
<?xml version="1.0"?>
<stream:stream xmlns:stream="http://etherx.jabber.org/streams"
xmlns="jabber:client" to="mydomain.com">
<message to=dns@mydomain.com from=correioelectronico@mydomain.com >
<subject>Alarm Message!!!</subject>
<body>
    <Inform>
        <language>fipa-s10</language>
        <ontology>AgentsManagement</ontology>
        <content>
            <error>Correio Electronico</error>
            <service>DNS</service>
            <description>Não resolve nomes.</description>
            <ticket>16</ticket>
            <notify>sgu@mydomain.com</notify>
        </content>
    </Inform>
</body>
</message>
</stream:stream>
```

Quando o agente DNS recebe o alarme, ele vai também iniciar um conjunto de procedimentos previamente programados com vista a resolver o problema (Figura 32 (1)), e nesses mesmos procedimentos verifica que o serviço de DNS não responde. O agente DNS inicia assim uma sessão remota no servidor de DNS para reiniciar o serviço, e o problema fica resolvido.

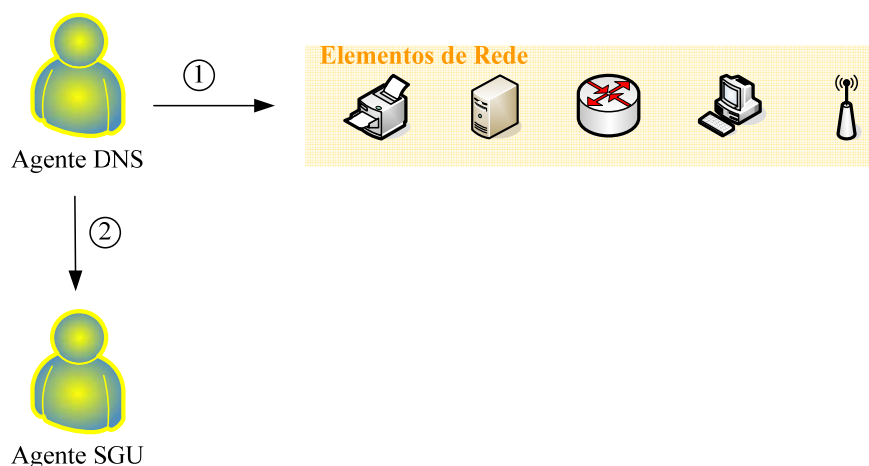


Figura 32 – Protótipo Inter Domínio 4.

Após a resolução do problema, o agente DNS notifica o agente SGU (Figura 32 (2)), responsável pela aplicação de gestão que gerou alarme, que o problema está resolvido de forma a fechar o *ticket* do alarme.

Tabela 17 – Notificação de problema resolvido para o agente SGU.

```
#Agente DNS notifica SGU de problema resolvido
<?xml version="1.0"?>
<stream:stream xmlns:stream="http://etherx.jabber.org/streams"
xmlns="jabber:client" to="mydomain.com">
<message to=sgu@mydomain.com from=dns@mydomain.com >
<subject>Alarm Message!!!</subject>
<body>
  <Inform>
    <language>fipa-s10</language>
    <ontology>AgentsManagement</ontology>
    <content>
      <error>Servico DNS</error>
      <service>DNS</service>
      <description>Problema resolvido.</description>
      <ticket>16</ticket>
    </content>
  </Inform>
</body>
</message>
</stream:stream>
```

5.1.2 Protótipo de utilização com Operador de Rede

Vamos assumir o protótipo de utilização autónomo anterior, mas neste caso o agente DNS não consegue contactar a máquina de DNS (Figura 33 (1)). Ele executou todos os procedimentos que tinha programado e mesmo assim não consegue aceder à máquina. O agente DNS decide então reiniciar a máquina, desligando a PDU onde o servidor está ligado (Figura 33 (2)). Uma vez que a partir daqui o agente DNS não tem informação sobre o estado da máquina, ele vê-se forçado a notificar o administrador ou operador de rede do que aconteceu (Figura 33 (3)).

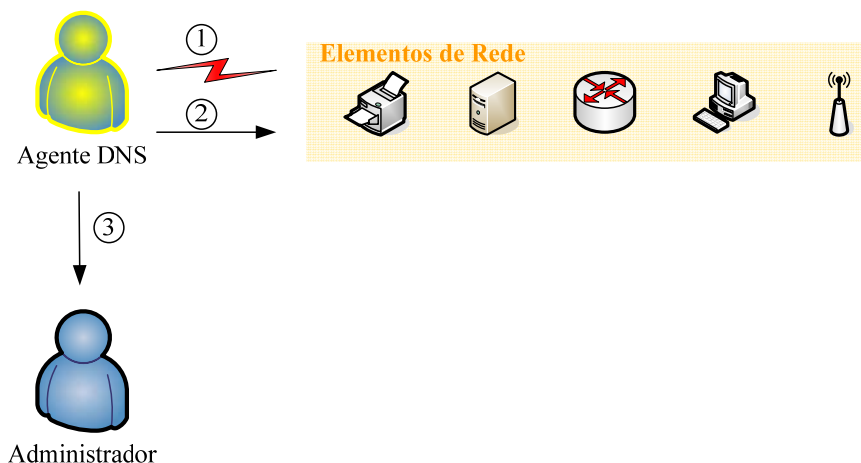


Figura 33 – Protótipo Inter Domínio com Administrador.

A notificação neste caso é feita também por *Instant Messaging*, tendo para isso o agente de DNS descobrir qual o contacto do administrador ou operador de rede.

Tabela 18 – Pesquisa de contacto do administrador.

```

#Pesquisa no LDAP
ldap_search_s(ld, "dc=mydomain,dc=com", 2,
(&(objectClass=person)(cn=Administrador)))" ,attrList,0,&msg)

# Resultado da Pesquisa
dn: cn=Administrador,cn=Users,dc=mydomain,dc=com
cn: Administrador;
displayname: Administrador;
mail: administrador@mydomain.com
description: Administrador ou Operador de Rede
  
```

Tabela 19 – Notificação ao Administrador.

```

Agente DNS notifica Administrador de reboot.
<?xml version="1.0"?>
<stream:stream xmlns:stream="http://etherx.jabber.org/streams"
xmlns="jabber:client" to="mydomain.com">
  
```



```
<message to=administrador@mydomain.com from=dns@mydomain.com >
<subject>Alarm Message!!!</subject>
<body>
  <Inform>
    <language>fipa-s10</language>
    <ontology>AgentsManagement</ontology>
    <content>
      <error>Servico DNS</error>
      <service>DNS</service>
      <description>Reinicie a maquina de DNS. Não respondia e tive
de forçar a desligar.</description>
      <notify>sgu@mydomain.com</notify>
      <ticket>16</ticket>
    </content>
  </Inform>
</body>
</message>
</stream:stream>
```

5.2 Ambiente de Gestão Inter Domínios

Como se pode gerir um recurso, ou serviço, se este depende de terceiros? Por exemplo o serviço de VoIP. A resposta tem sido frequentemente dada com base em cenários e probabilidades.

Num ambiente de rede em que o aumento das dimensões e fundamentalmente da dispersão geográfica da mesma é uma realidade cada vez mais frequente, os aspectos da gestão, envolvendo o recurso a terceiros (redes de fornecedores de serviços de comunicação, redes de terceiros, etc.) são cada vez mais críticos.

O ambiente de gestão intra-domínio é composto por uma rede privada e com um fornecedor de serviço (ISP) conforme se pode ver na figura seguinte.

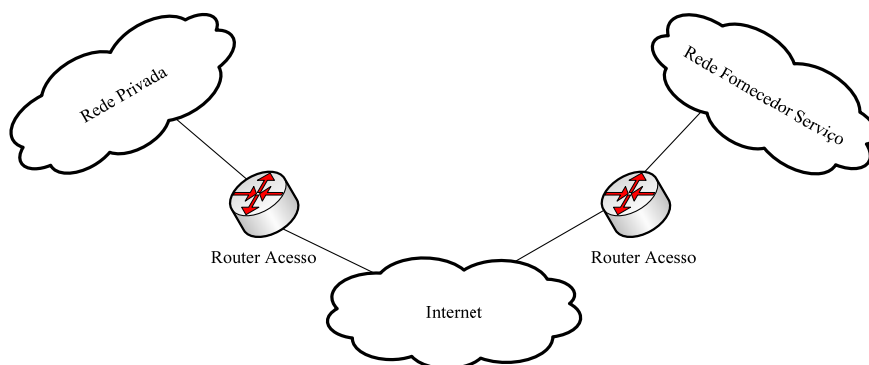


Figura 34 – Ambiente de Gestão Inter Domínio.

Como será de esperar, o administrador ou operador de rede pretende ter todas as garantias e mais que isso, poder monitorizar, delegar e intervir em caso de algo não estar a ocorrer como programado.

Para tal todas as aplicações de gestão mencionadas acima no ambiente de gestão Intra-Domínio estão presentes aqui também, quer no lado da rede privada, quer na rede de operador.

5.2.1 Protótipo de utilização Inter-Domínio

Neste caso de utilização, a aplicação responsável por gerir o SLA contratado da Internet verifica que os valores contratados não são iguais aos valores fornecidos.

Esta situação torna-se particularmente grave para os serviços que dependem da Internet, como por exemplo o serviço de VoIP, que pode impedir a realização de chamadas para o exterior.

A aplicação que gere os SLA's, gera um alarme a alertar que o SLA contratado não é igual ao fornecido, e envia um e-mail para o administrador ou operador de rede através de uma interface IMAP.

Aqui a Aplicação Autónoma *Dispatcher Agent*, recebe o alarme gerado pela aplicação de gestão, uma vez que tem mapeada uma conta IMAP nas aplicações de gestão existentes.

Quando o *Dispatcher Agent* recebe o alarme (Figura 35 (1)), vai analisá-lo para tentar saber quem é responsável pela sua resolução, de forma a lhe dar seguimento.

O *Dispatcher Agent* verifica que o problema é com a Internet e que vai ter de despachar o alarme para o fornecedor de serviço. Para saber como contactar o fornecedor de serviço, o “*Dispatcher Agent*” faz uma pesquisa no DNS (Figura 35 (1)) pelo serviço de LDAP no domínio do fornecedor de serviço. A partir daqui, como o endereço do LDAP no domínio do fornecedor de serviço já é conhecido, o *Dispatcher Agent* vai fazer uma pesquisa no LDAP (Figura 35 (2)) pelo agente responsável pela abertura de *tickets* por parte dos clientes.

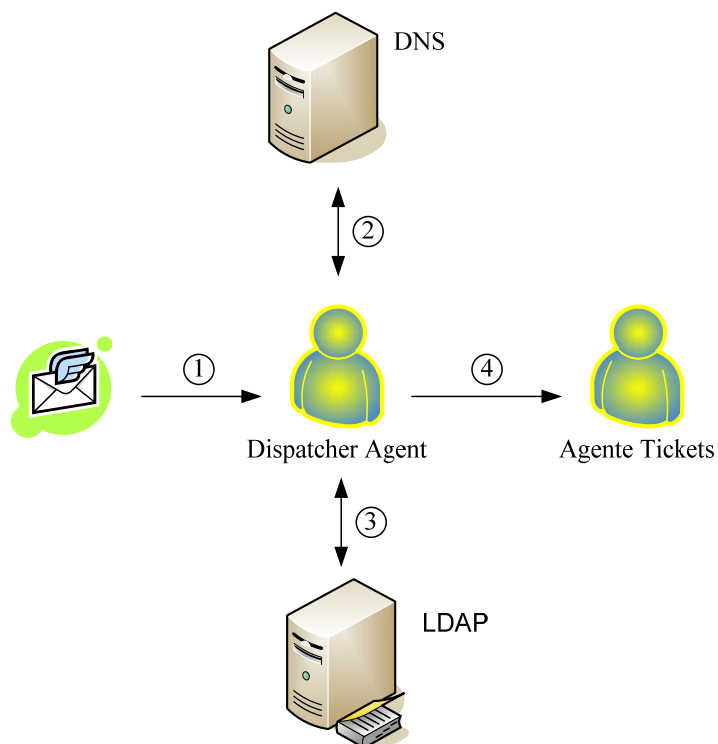


Figura 35 – Protótipo Intra-Domínio.

Tabela 20 – Pesquisa/Resultado por Agente/Serviço de *Tickets* no LDAP.**#Pesquisa no LDAP**

```
ldap_search_s(ld, "ou=Agentes,ou=agency,dc=otherdomain,dc=com", 2,
(&(objectClass=agent)(service-name=TicketAgent)))" ,attrList,0,&msg)
```

Resultado da Pesquisa

```
dn: cn= TicketAgent,ou=Agentes,ou=agency,dc=otherdomain,dc=com
cn: TicketAgent;
displayname: Agente Ticket;
mail: ticketagent@otherdomain.com
description: Agente Abertura Tickets
```

Assim que o agente/serviço é encontrado, o *Dispatcher Agent* vai iniciar uma sessão de *Instant Messaging* com o *TicketAgent*, que conforme o seu estado de presença, vai passar a informação do alarme via *Instant Messaging* (Figura 35 (4)), de forma a dar seguimento à resolução do problema.

Tabela 21 – Notificação do Operador.

#Dispatcher envia dados ao TicketAgent.

```
<?xml version="1.0"?>
<stream:stream xmlns:stream="http://etherx.jabber.org/streams"
xmlns="jabber:client" to="mydomain.com">
```

```
<message to=ticketagent@otherdomain.com from=dispatcher@mydomain.com >
<subject>Alarm Message!!!</subject>
<body>
  <Inform>
    <language>fipa-s10</language>
    <ontology>AgentsManagement</ontology>
    <content>
      <error>Servico Internet</error>
      <service>Internet</service>
      <description>SLA for a parâmetros contratados. Internet
lenta.</description>
      <notify>admin@mydomain.com</notify>
      <ticket>16</ticket>
    </content>
  </Inform>
</body>
</message>
</stream:stream>
```

A partir daqui o *TicketAgent* contém os dados do alarme do cliente e vai abrir um *ticket* no sistema de *HelpDesk* do operador de forma a dar seguimento ao alarme. Após o *ticket* ter sido criado, o processo de resolução do mesmo é igual ou semelhante mencionado acima no ambiente de Inter Domínio.

5.3 Conclusões

Conforme se pode verificar, a resolução de problemas é mais rápida, quase em tempo real devido ao uso de *Instant Messaging* e também devido ao uso de Aplicações Autónomas, uma vez que estas recebem os alarmes e dão seguimento à sua resolução, uma vez que o acesso à informação de gestão é feita de forma aberta e normalizada e também devido uso do *Instant Messaging*.

Uma Aplicação Autónoma pode actuar directamente nos elementos de rede, através do uso de protocolos de gestão como o SNMP/CMIP ou então iniciar uma sessão num elemento responsável por um determinado serviço e tentar colocar o serviço a funcionar.

No protótipo apresentado não implica a presença física do operador, salvo algumas excepções como por exemplo problemas em cablagens, etc.

Por outro lado se a Aplicação Autónoma não for capaz de resolver o incidente sozinha, ela pode pesquisar no servidor de LDAP por outra Aplicação Autónoma capaz de a ajudar a resolver o problema, através da decomposição do problema em vários pequenos problemas por exemplo.

De forma análoga, o administrador ou operador de rede pode monitorar o que os agentes estão a fazer, trocando mensagens com eles através de *Instant Messaging*. No que diz respeito à notificação de alarmes por parte das aplicações de gestão, fica tudo na mesma, acontece é que existem mais receptores.

Tudo isto é possível de forma simples. Todos os protagonistas da gestão têm de ter uma forma normalizada de comunicação, a nossa proposta assenta em *Queries LDAP* e *Instant Messaging*, e também uma forma de partilharem a sua informação de gestão, de forma a ser possível às aplicações de gestão partilharem a sua informação nos diferentes níveis de gestão.

Capítulo 6

6 Conclusões

Esta Tese, tinha como objectivo, a integração do *Instant Messaging* na Arquitectura de Gestão, trabalho esse que se dividiu em quatro áreas principais como a Arquitectura de Gestão, o uso de LDAP, o *Instant Messaging* e finalmente a linguagem de conteúdo.

A Arquitectura de Gestão, passou a dar respostas efectivas e a viabilizar situações, que através do recurso aos sistemas de gestão tradicionais seriam impossíveis, tais como a gestão remota, comunicação em tempo real entre sistemas gestores, mesmo dispersos geograficamente e partilha de informação de gestão.

Desta forma, a gestão integral e integrada multi sistemas num domínio de rede, a gestão inter-domínio e a gestão partilhada de recursos inter-domínio, passam assim, a ser funcionalidades que tornam a arquitectura aqui apresentada como uma das revelações para a gestão de redes em ambiente aberto.

A separação da consola do sistema gestor e o recurso a um ambiente gráfico através de consolas "*browser oriented*", vem permitir, aos administradores ou operadores de rede, homogeneizar os procedimentos, bem como a forma de utilização da informação de gestão. Assim, é possível a utilização da mesma consola, para monitorizar e operar múltiplos sistemas gestores, além de aumentar a mobilidade dos operadores de gestão.

Por outro lado, a capacidade de comunicação, através do uso de várias interfaces e protocolos de Internet já existentes, fornece ao operador a consola mais adequada, como por exemplo, um *browser* de computador, um telefone móvel, PDA, ou outro equipamento.

A utilização de sistemas de directórios, em particular a utilização do LDAP, que é o serviço de directório mais adoptado para o acesso a este tipo de sistemas, é uma característica fundamental deste trabalho.

Como protocolo normalizado, o LDAP, tem um papel essencial na integração de Aplicações de Gestão e Aplicações Autónomas, num ambiente de gestão de redes intra e inter-domínio. O Modelo Global de Informação desenvolvido, permite assim a partilha e integração da informação de gestão, desde o nível dos elementos até ao nível de negócio.

Surge assim a possibilidade de leitura da mesma informação, quer por Utilizadores humanos, quer por Aplicações Autónomas, permitindo evoluir no sentido de uma maior automatização e supervisão humana, uma vez que a forma de aceder a esta informação é fácil, aberta e normalizada.

Assim, através do uso dos sistemas de directório e do LDAP para lhes aceder, as Aplicações Autónomas, podem publicar a sua informação no servidor de LDAP, partilhando desta forma a sua informação com Utilizadores humanos e outras Aplicações Autónomas.

Os dados publicados pelas Aplicações Autónomas, podem incluir informações como serviços, descrições, linguagens suportadas, ontologias, estado do agente, entre outras.

A Arquitectura apresentada, vem permitir a comunicação *on-line* entre Utilizadores humanos e Aplicações Autónomas, através do uso de Sistemas de *Instant Messaging*, e do protocolo XMPP,.

O aspecto mais importante do *Instant Messaging*, que o diferencia de outros sistemas de comunicação, é o facto de integrar informação de presença, fornecendo a capacidade de monitorizar o estado de Utilizadores e Aplicações Autónomas na rede.

Existe ainda uma tendência, para o aumento do uso de sistemas que utilizem o efeito presença, não só ao nível de Utilizadores humanos, mas também ao nível das Aplicações Autónomas, permitindo assim alertas e notificações automáticas através do uso de interfaces *Instant Messaging*.

A utilização do XML como linguagem de comunicação, permite separar a informação de gestão da sua apresentação.

Esta característica, acrescenta a esta Arquitectura, uma outra mais valia, a capacidade de observar a informação disponível, de acordo com o seu padrão de apresentação definido e com o dispositivo disponível.

É a capacidade de independência da informação, face à sua forma de apresentação, que a utilização do XML incorpora mais valias nesta Arquitectura.

Uma outra mais valia, não menos importante que as anteriores, é o facto, de através da utilização do XMPP, XML e XSL, ser possível permitir que, quer humanos quer Aplicações Autónomas, possam utilizar e trocar a mesma informação.

Se a informação for utilizada por um operador, através de uma consola, este associa o XML ou XSL respectivo e consegue assim observar e analisar a informação de uma forma gráfica.

Por outro lado, no caso de se tratar de uma Aplicação Autónoma a fazer uso da mesma informação, esta utiliza unicamente, a informação transportada em XML e encontra-se assim em condições de executar os procedimentos que achar mais apropriados de forma a solucionar o problema.

6.1 Trabalho Futuro

O futuro deste trabalho, passa pela melhoria da linguagem de conteúdo para tornar mais simples a interacção humana, bem como das ontologias, de forma a permitir às

Aplicações Autónomas comunicarem entre si, bem como interagirem com Utilizadores humanos.

A utilização de *Buddy Lists*, também deve ser objecto de estudo futuro, uma vez que se podem criar grupos no *Instant Messaging*, de Aplicações Autónomas especializadas, por exemplo, nos diferentes níveis de gestão ou então por ontologias.

Um outro aspecto importante, para o futuro deste trabalho, passa pelo desenvolvimento de um método, que permita a conversão de linguagem natural, utilizada pelos Utilizadores humanos, em XML, utilizada por Aplicações Autónomas, e vice-versa. Com este mecanismo, a interacção humana ficaria bastante mais facilitada, na medida em que, o Utilizador humano, escrevia as mensagens normalmente num interface gráfico e que depois, as mensagens eram convertidas para XML para serem interpretadas pelas Aplicações Autónomas. A interacção Homem – “Máquina” ficaria assim mais facilitada.

Bibliografia

- [1] Raul Filipe Teixeira de Oliveira. Managing Awareness in Networks Through Software Agents. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 1998.
- [2] Alex L. G. Hayzelden and John Bigham. Future communication networks using software agents. In Alex L. G. Hayzelden and John Bigham, editors, Software Agents for Future Communication Systems, chapter 1, pages 1{57. Springer, March 1999.
- [3] Decina M. and Trecordi V. Convergence of telecommunications and computing to networking models for integrated services and applications. In IEEE, volume 85, number 12, December 1997.
- [4] Goldszmidt G. and Yemini U. Delegated agents for network management. IEEE Communications Magazine, 36, number 3, March 1998.
- [5] S. Green, L. Hurst, B. Nangle, P. Cunningham, F. Sommers, and R. Evans. Software agents: A review. Technical report, Trinity College, University of Dublin, June 1997. Available at <http://www.cs.tcd.ie=research;groups=aig=iag>
- [6] Michael J. Wooldridge and Nicholas R. Jennings., Agent Theories, Architectures, and Languages: A Survey , Knowledge Engineering Review Vol. 10 N°2 - Cambridge University Press, Julho 1996.
- [7] Rodney A. Brooks, Intelligence Without Reason , A.I. Memo No. 1293 April, 1991, Prepared for Computers and Thought, IJCAI-91, 1991.
- [8] FIPA Agent Management Specification, SC00023K, 2004 <http://www.fipa.org/specs/fipa00023/index.html>
- [9] FIPA Agent Message Transport Protocol for IIOP Specification, SC00075G, 2002 <http://www.fipa.org/specs/fipa00075/index.html>
- [10] FIPA Agent Message Transport Protocol for HTTP Specification, SC00084F, 2002 <http://www.fipa.org/specs/fipa00084/index.html>
- [11] Michael R. Genesereth and Steven P. Ketchpel. Software agents. Communications of the ACM, 37(7), July 1994.

- [12] Klaus Fischer, Jorg P. Muller, and Markus Pischel. A pragmatic bdi architecture. In Michael Wooldridge, Jorg P. Muller, , and Milind Tambe, editors, *Intelligent Agents II – Agent Theories, Architectures, and Languages*, pages 203-218. Springer, 1995.
- [13] Cohen, P. R., and Levesque, H. J. Performatives in a rationally based speech act theory. In *Proceedings of the 28 th Annual Meeting of the Association for Computational Linguistics*, Pittsburgh, Pennsylvania. 1990b.
- [14] FIPA ACL Message Structure Specification, SC00061G, 2002 <http://www.fipa.org/specs/fipa00061/sc00061G.html>
- [15] Tim Finin, Yannis Labrou, and James Mayfield. Kqml as an agent communication language. In *Software Agents*, chapter 14, pages 291{316. AAAI Press/The MIT Press, 1997.
- [16] M Wooldridge and N. R. Jennings. *Intelligent agents: Theory and practice*. The Knowledge Engineering Review, 1995.
- [17] J. P. MÄuller. Architectures and applications of intelligent agents: A survey. The Knowledge Engineering Review, 1998.
- [18] Mães, P. Agents that reduce Work and Information Overload. Em Rieckem, d. (editor). *Tema especial: Intelligent Agents Communications of the ACM*, 37(7) Julho 1994.
- [19] Milojicic, D., Douglis, F.e Wheeler, R. (editores). *Mobility: Processes, Computers, and agents*. Addison-Wesley / ACM Press. February, 1999.
- [20] Rodrigues da Silva, A., Romão, A., Deugo, D., e Mira da Silva, M. Towards a Reference Model for Surveying Mobile Agent Systems. *Journal of Autonomous Agents and Multi-Agent Systems*, Kluwer Academic Press. 2000.
- [21] Phil Shafer. XML – based network management. Juniper Networks white paper, 2001.
- [22] Lightweight Directory Access Protocol (v3), UTF-8 String Representation of Distinguished Names, RFC 2253.
- [23] Hani Mansi Harri Levanen, Bernard Freund. Using LDAP for directory integration – a look at ibm secureway directory, active directory and domino. IBM Redbooks, December, 2000.
- [24] Harri Stranden Heinz Johner, Michel Melot. LDAP implementation cookbook. IBM Redbooks, June 1999.
- [25] Raul Oliveira Bruno Marques, Edgar Nogueira. LDAP role in network management. HPOVUA 2004, May, 2004.
- [26] José A. Folha, Bruno F. Marques, José A. Oliveira, Paulo M. Coelho, Raul F. Oliveira, Instant Management Infrastructure Through Network Management Support Systems and Software Agents Coordination, ICETE/ICE-B 2006, August 2006.

- [27] Bruno F. Marques, Edgar Nogueira, Jose A. Oliveira, and Raul F. Oliveira, Ldap role in network management, 11th HP OpenView University Associations Workshop, Paris, June 2004.
- [28] FIPA Abstract Architecture Specification, SC00001L, 2004
<http://www.fipa.org/specs/fipa00001/index.html>
- [29] Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence RFC 3921.
- [30] Jabber Software Foundation, <http://www.jabber.org>
- [31] Thomas R. Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In N. Guarino and R. Poli, editors, Formal Ontology in Conceptual Analysis and Knowledge Representation, Deventer, The Netherlands, 1993. Kluwer Academic Publishers.
- [32] Nicola Guarino and P. Giaretta. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In N. Mars, editor, Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, pages 25–32. Ed. Amsterdam: ISO Press, 1995.
- [33] Ann Wrightson. Topic Maps and Knowledge Representation. Ontopia, February, 2001. <http://www.ontopia.net/topicmaps/materials/kr-tm.html>.
- [34] FIPA ACL Message Representation in XML Specification, SC00071e, 2002
<http://www.fipa.org/specs/fipa00071/sc00071e.pdf>
- [35] Messaging Pipeline
<http://www.messagingpipeline.com/showArticle.jhtml?articleID=51000335>

